





Article

A Robust Robotic Disassembly Sequence Design Using Orthogonal Arrays and Task Allocation

Mohammad Alshibli ¹, Ahmed ElSayed ^{1,*}, Elif Kongar ², Tarek Sobh ¹ and Surendra M. Gupta ³

¹ Department of Computer Science and Engineering, School of Engineering, University of Bridgeport, 221 University Avenue, Bridgeport, CT 06604, USA; malshibl@my.bridgeport.edu (M.A.); sobh@bridgeport.edu (T.S.)

² Department of Technology Management, School of Engineering, University of Bridgeport, 221 University Avenue, Bridgeport, CT 06604, USA; kongar@bridgeport.edu

³ Department of Mechanical and Industrial Engineering, College of Engineering, Northeastern University, 360 Huntington Avenue, Boston, MA 02115, USA; S.Gupta@northeastern.edu

* Correspondence: aelsayed@my.bridgeport.edu

Received: 20 December 2018; Accepted: 7 March 2019; Published: 11 March 2019



Abstract: Disassembly sequence planning (DSP) is a nondeterministic polynomial time (NP) complete problem, making the utilization of metaheuristic approaches a viable alternative. DSP aims at creating efficient algorithms for deriving the optimum or near-optimum disassembly sequence for a given product or a product family. The problem-specific nature of such algorithms, however, requires these solutions to be validated, proving their versatility in accommodating substantial variations in the problem environment. To achieve this goal, this paper utilizes Taguchi's orthogonal arrays to test the robustness of a previously-proposed Simulated Annealing (SA) algorithm. A comparison with an exhaustive search is also conducted to verify the efficiency of the algorithm in generating an optimum or near-optimum disassembly sequence for a given product. In order to further improve the solution, a distributed task allocation technique is also introduced into the model environment to accommodate multiple robot arms.

Keywords: disassembly sequence planning; electronic product recovery; industrial robots; recycling; robotic manipulation; simulated annealing; task allocation

1. Introduction

According to the global e-waste monitor, 44.7 million metric tons of electronic waste (e-waste) was generated globally in 2016 (Figure 1), with an estimated 6.1 kg e-waste per habitant [1]. As depicted in Figure 1, e-waste has been growing steadily over the past several years, and this trend is expected to continue. The utilization and subsequent re-utilization of recyclable materials and reusable components is often cited as the most viable solution to reducing user waste.

To achieve this, several industries today are adopting varying levels of take-back policies in an attempt to regain the materials and component content embedded in products that have completed their useful lives. End-of-life (EOL) management systems, following a systematic operational flow to regain the value embedded in e-waste, deal with creating economically-, environmentally-, and socially-benign solutions. The first step in EOL management systems involves the collection of EOL products from various warehouse and landfill locations. Following the required sorting and inspection operations, the second step is primarily concerned with regaining the value in EOL products via EOL processes, such as recycling or reuse, or, alternatively, storing the components for possible future use or properly disposing of them to minimize the environmental hazard.

The majority of EOL processing operations require a certain level of disassembly. Specifically, electromechanical products with modular product structures are made up of a large number of components. This requires an optimal disassembly sequence to ensure the efficiency of overall EOL operations. Despite the fact that exhaustive search algorithms are capable of achieving this task, the combinatorial nature of the problem makes the path selection cost and time prohibitive. To address this issue, heuristic methods are often employed to obtain near-optimal or optimal solutions.

Adding to the complexity of the overall problem environment, compared to conventional mass manufacturing/remanufacturing operations, EOL product recovery systems have to deal with significantly higher levels of product variety and vagueness due to the uncertainties of the input stream. Therefore, disassembly lines should be designed to accommodate a large variety of products with imprecise bill of material (BOM) data. With this motivation, this study builds on an environmentally-benign and economically-feasible disassembly sequencing approach, and proposes an automated multi-resource robotic disassembly platform. A robustness analysis using Taguchi orthogonal array (OA) design is conducted to demonstrate the flexibility of the algorithm. Exhaustive search comparison is also provided to verify the efficiency of the proposed solution.

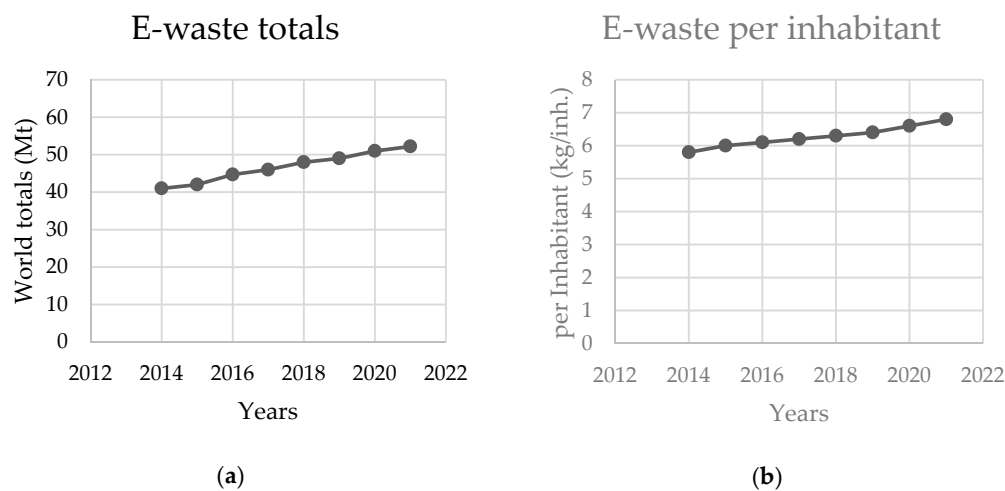


Figure 1. Total E-waste (a) and E-waste per inhabitant (b) for years 2014–2021 (values from 2017 to 2021 are estimated).

This paper is organized as follows: The literature review, highlighting previous research on related work, is provided in Section 2. This section is comprised of two parts, and discusses previously-published studies on robustness design and disassembly line balancing. Detailed descriptions of the problem and the building blocks of the proposed methodology are presented in Section 3. Section 4 provides data regarding the practical application of the proposed methodology for task allocation and distribution using a personal computer as an example. Concluding remarks and the implications for future research are given in Section 5.

2. Literature Review

Disassembly sequencing is considered to be an NP-complete problem [2,3]. As with all NP-complete problems, the complexity of the problem increases exponentially with the number of components in a given product's structure, justifying the utilization of metaheuristic methods.

Many researchers have investigated and implemented algorithms to solve the disassembly sequence problem, e.g., [4–9]. The quality of these models relies on their versatility, viz., their ability to provide reliable solutions to a wide range of problems in an efficient and effective manner. That is, the proposed algorithms should stay loyal to the bill of materials and should be able to preserve precedence relationships in the given product structure. These algorithms are designed to provide near-optimum/optimum solutions to any combinatorial optimization problem including the

disassembly sequences generation problem. Such algorithms provide viable solutions relatively more quickly, regardless of the deviations in the numerical data introduced to the model environment [10]. The Taguchi Orthogonal Array (OA) design and Design of Experiments (DOE) are two methods that are commonly used for this purpose.

Experimental design was first introduced in the 1920s by R. A. Fischer, who developed the basic principles of factorial design and the associated data analysis known as ANOVA [11]. DOE has gained wide interest and applications, especially in the field of engineering and science in optimization, process management, and development. DOE is an experimental method that is used to signify the relationship between input parameters and the output results statistically [12,13].

Taguchi Orthogonal Array (OA) and DOE with meta-heuristic optimization have been utilized to validate the robustness of the experiment in terms of its applicability to a large variety of products and industries [12,14,15].

In the manufacturing arena, Kondapalli et al. [12] have reviewed the literature on DOE techniques that have been employed for different welding processes. The authors also focused on the application of the Taguchi method on fusion arc welding processes.

For most disassembly systems, there are two crucial issues which need to be considered: (1) disassembly sequence generation, which looks into determining the optimal or near-optimal disassembly sequence, and (2) disassembly-to-order (DTO) system optimization, which deals with obtaining the optimal numbers of end of life (EOL) products to be processed to fulfill the demand for components and materials while achieving multiple goals and objectives.

Addressing these two issues, Ilgin and Tasoglu [14] utilized an orthogonal array experimental design, a statistical technique that systematically tests the proposed algorithm when dealing with a limited number of inputs. The authors have proposed a simulation optimization approach based on a genetic algorithm (GA) for the simultaneous determination of disassembly sequence and disassembly-to-order decisions. The authors illustrated their proposed approach using a numerical example, and carried out a Taguchi's L9 orthogonal array experimental design to obtain the best values of GA parameters. This orthogonal array included four factors with three levels. Similarly, El-Sayed et al. [5] presented a genetic algorithm model to obtain the optimal disassembly sequence of a given product in which precedence relationships were preserved.

Chang [16] presented a method that combines a particle-swarm optimization with nonlinear time-varying evolution and orthogonal arrays (PSO-NTVEOA) in the planning of harmonic filters for the high-speed railway traction system. The paper aimed to minimize the cost of the filter, the filter losses, and the total harmonic distortion of currents and voltages at each bus, simultaneously.

Disassembly line balancing (DLB) is known as the arranging of a group of tasks into an ordered sequence of stations for performance optimization. DLB problems, while pursuing disassembly line optimization, also aim to meet the demand for the parts retrieved from the returned products [17]. Several steps of recovery and remanufacturing are included in the disassembly process. Aiming to solve DLB problems through resource constraints, Metea et al. [18] provided a mathematical model to minimize the number of resources and workstations.

Bentaha et al. [19] considered disassembly line balancing and sequencing problems in the presence of hazardous parts of the EOL product. The authors aimed at proposing a production line that generates a maximum profit under the uncertainty of task times.

Disassembly line balancing is one of the most efficient techniques to accomplish efficiency in the disassembly of large quantities of returned products. With this motivation, Güngör et al. [17] studied the main challenges and problems in disassembly line balancing, and presented a heuristic model to demonstrate the impact of several important factors in disassembly operations. Similarly, Gagnon and Morgan [20] carried a review of the documented decisions and issues, focusing on the complications in disassembly line balancing problems.

The disassembly process is subject to interruptions due to defects and machine breakdowns, leading to complications and delays. Addressing similar issues, Gungor and Gupta [21] discussed

the disassembly line balancing problem in the presence of task failures (DLBP-F), and highlighted the complications that may occur.

In the area of automated disassembly, Torres et al. [22] proposed a non-destructive automatic disassembly of personal computers. The authors employed a computer vision system to recognize and localize the product and its components. Gutjahr and Nemhauser [23] presented a solution to the assembly line balancing problem, minimizing the delays at each workstation based on the shortest route in a finite, directed network. A similar study has been proposed by Erel and Gokcen [24], which allowed mixed-model lines that incorporated multiple state times before categorizing a given set of completed tasks to a workstation.

McMullen and Tarasewich [25] used ant colony optimization to solve the assembly-line balancing problem with parallel workstations, stochastic task durations, and mixed models.

The disassembly line balancing problem searches for a feasible sequence to minimize the number of workstations by reducing idle times. To achieve this, McGovern and Gupta [2] presented a genetic algorithm to obtain optimal or near-optimal solutions for disassembly line balancing problems.

Duta et al. [26] designed and balanced a disassembly line through a method which relies on the equal piles approach to avoid uncertainties that occur during the disassembly process. Furthermore, Duta and Filip [27] studied the line structure and proposed an algorithm that aimed at finding the best disassembly sequence.

Using a preference ranking organization method for the enrichment of evaluations (PROMETHEE), Avikal et al. [28] proposed an efficient, near-optimal, and a multi-criteria decision making technique-based heuristic for assigning disassembly tasks to available workstations.

The majority of these line balancing algorithms are computationally complex, requiring platforms that will limit the restrictions on the computational power. With technological advances, serverless computing is gaining considerable interest and has been adopted by several organizations due to its powerful services, simple programming, and deployment models. Serverless runtime allocates resources as events arrive, which can avoid the high costs of pre-allocated or dedicated hardware. However, the serverless architecture models are still in their infancy, and the literature survey indicates that they are not considered yet for automated disassembly systems where multiple robot arms are present.

One relevant work has been proposed by McGrath and Brenner [29], in which the authors presented a serverless computing platform implemented in .NET. Their findings indicated that the prototype achieved better throughput than other platforms at most concurrency levels. Ho and Lee [30] presented a new data reorganization algorithm that allowed a controllable tradeoff between data reorganization overhead and streaming load balance.

Bolosky et al. [31] calculated the results on disk usage, content, and file activity, and also considered machine uptimes, lifetimes, and loads. They concluded that the measured desktop infrastructure would possibly support their proposed system, providing availability about one unfilled file request per user per thousand days. Hendrickson et al. [32] proposed a new, open-source platform for building next-generation web services and applications in the burgeoning model of serverless computation called OpenLambda. A brief summary of the current web applications was also included in order to motivate some aspects of serverless applications. Bila et al. [33] presented a serverless architecture for securing Linux containers, which provides continuous scanning. The authors explored the design of an automated threat mitigation architecture based on OpenWhisk and Kubernetes. Baldini et al. [34] introduced the serverless trilemma, which captures the inherent tension between economics, performance, and synchronous composition. Using serverless distributed file system, Adya et al. [35] improved the performance of file storage.

This study builds on the disassembly sequence problem investigated in [4], and validates the robustness of the heuristic and metaheuristic algorithms employed to solve the disassembly sequence generation and line balancing problems in a multi-arm robotic system. To this end, a test and performance evaluation for the proposed Simulated Annealing (SA) disassembly sequence problem

using Taguchi orthogonal arrays (OA) is presented. Following this, a hypothesis test over the mean runtime is conducted to statistically prove the effectiveness of the SA algorithm. In addition, the SA model is enhanced to comprise multiple symmetric robot arms. The algorithm aims at obtaining the optimum or near-optimum task allocation among available robot arms while deploying all to ensure the efficiency of their utilization.

An additional improvement introduced in this work involves resource allocation. The majority of optimization problems require adequate resources to be allocated to the metaheuristic algorithm for faster execution times while generating the optimum or near-optimum solution(s). This is especially true when personal devices with limited capacities are utilized. To address this issue, the authors recommend an architecture that would help the algorithm acquire the required resources to generate the solution in a more efficient manner. For further information regarding multiple criteria decision making applications in environmentally conscious manufacturing and product recovery, see [36].

3. Robustness Design Using Orthogonal Arrays

This section analyzes the robustness of the Simulated Annealing algorithm proposed by Alshibli et al. [4] using Orthogonal Arrays (OAs) [37]. A detailed explanation of the SA implementation is also included in this section.

Table 1 and Figure 2 represent the product utilized in [4], viz., the PC under disassembly, its components, their material content, the disassembly method required for each component, the hierarchy and the dependencies amongst the components in the end-of-life (EOL) product.

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

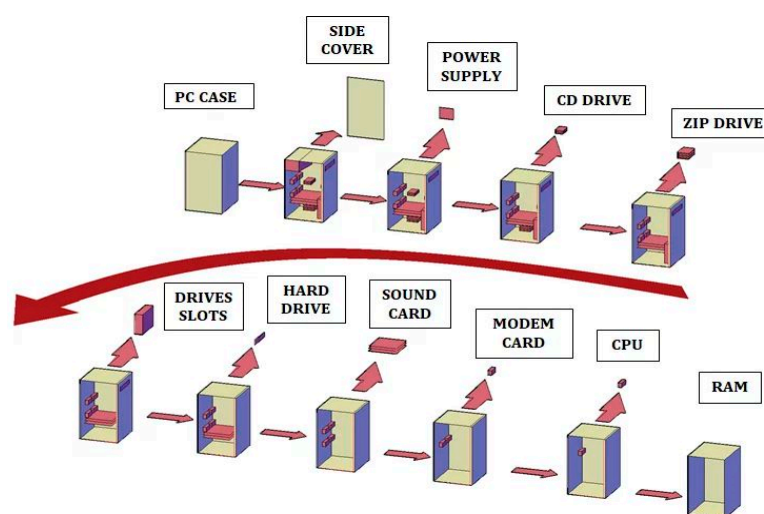


Figure 2. EOL product component hierarchy.

As can be seen from the figure, for this sample, there are 10 components in the EOL product structure. For the type of materials of components and other product information, see Alshibli et al. [4]. The algorithm proposed in this study utilized constant disassembly times for each component, in addition to a predefined and constant speed for the robot arm and a penalty for the disassembly method change. Using these values, an optimum disassembly sequence was reached for this given EOL product.

In order to generalize the findings, orthogonal arrays are used to test the robustness of the proposed disassembly sequence generation algorithm. Here, the disassembly time is considered to be normally distributed, with varying values of the mean (μ) and standard deviation (σ) for the EOL product with the ten components provided in Figure 2.

In addition, the robot speed and the time required for the disassembly method change, e.g., from non-destructive to destructive, are also assumed to be normally distributed, creating the need for a 24-variable orthogonal array. With three levels for each variable, the full factorial would translate to $3^{24} = 282,429,536,481$ experiments. Using orthogonal arrays, the number of experiments are reduced to 54 as shown in Table 2 [38].

Table 2. Partial view of the reduced orthogonal array (OA), L54 ($2^1 \times 3^{25}$) [37,38].

Expt. No.	L54 (2 ¹ × 3 ²⁵) Orthogonal Array Column													
	μ(dt _{1j})	σ(dt _{1j})	μ(dt _{2j})	σ(dt _{2j})	...	μ(dt _{9j})	σ(dt _{9j})	μ(dt _{10j})	σ(dt _{10j})	μ(sf)	σ(sf)	μ(mt _{ij})	σ(mt _{ij})	
1	1	1	1	1		1	1	1	1	1	1	1	1	
2	1	1	1	1		2	2	2	2	2	2	2	2	
3	1	1	1	1		3	3	3	3	3	3	3	3	
4	1	2	2	2		3	2	3	2	3	2	3	2	
5	1	2	2	2	.	1	3	1	3	1	3	1	3	
6	1	2	2	2	.	2	1	2	1	2	1	2	1	
7	1	3	3	3	.	2	3	2	3	2	3	2	3	
8	1	3	3	3		3	1	3	1	3	1	3	1	
9	1	3	3	3		1	2	1	2	1	2	1	2	
10	2	1	1	2		1	2	3	2	3	2	3	2	
...					
44	2	3	1	2		2	3	1	1	3	3	1	2	
45	2	3	1	2		3	1	2	2	1	1	2	3	
46	3	1	3	2		2	2	3	3	2	1	1	2	
47	3	1	3	2		3	3	1	1	3	2	2	3	
48	3	1	3	2	.	1	1	2	2	1	3	3	1	
49	3	2	1	3	.	1	3	2	1	1	2	3	3	
50	3	2	1	3	.	2	1	3	2	2	3	1	1	
51	3	2	1	3		3	2	1	3	3	1	2	2	
52	3	3	2	1		3	1	1	2	3	3	2	1	
53	3	3	2	1		1	2	2	3	1	1	3	2	
54	3	3	2	1	...	2	3	3	1	2	2	1	3	

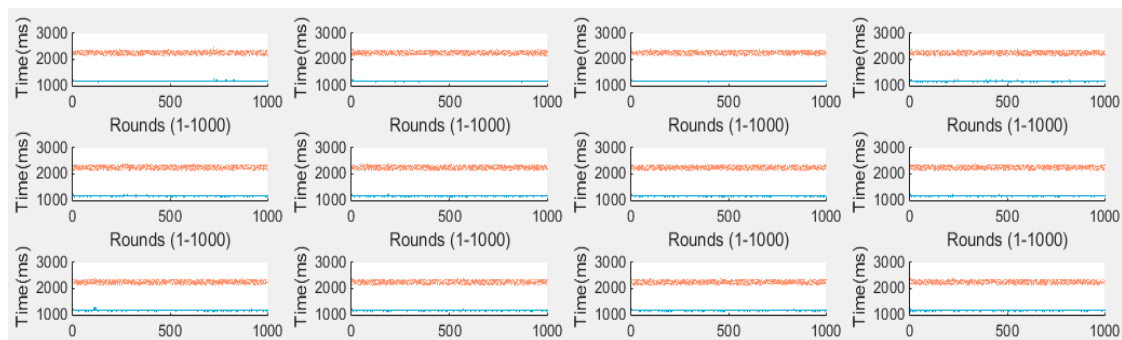
For each parameter generated in the set of means and standard deviations, the three levels are represented in Table 2 [37,38]. The levels 1, 2 and 3 are replaced by the actual values generated using the mean and standard deviation in Table 3.

Table 3 represents the value of each parameter for every experiment conducted. Here, column 1 represents the number of the experiment, columns 2 to 21 are the averages and standard deviations of disassembly times for each component, columns 22 and 23 represent the average and standard deviation of the robot speed, and columns 24 and 25 represent the average and the standard deviation of the disassembly method change, respectively.

For further analysis, each set of data was run 1000 times using both exhaustive search (marked in red) and the SA method proposed in [4]. Figure 3 depicts a partial view of the Simulated Annealing (SA) (marked in blue), and Exhaustive Search (ES) (marked in red) run times of 54 experiments.

Table 3. Partial view of the experimental data used in disassembly sequencing based on L54 ($2^7 1 \times 3^2 5$) [37,38].

Expt. No.	L54 ($2^7 1 \times 3^2 5$) Orthogonal Array Column											
	$\mu(dt_{1j})$	$\sigma(dt_{1j})$	$\mu(dt_{2j})$	$\sigma(dt_{2j})$...	$\sigma(dt_{9j})$	$\mu(dt_{10j})$	$\sigma(dt_{10j})$	$\mu(sf)$	$\sigma(sf)$	$\mu(mt_{ij})$	$\sigma(mt_{ij})$
1	2.000	0.010	3.000	0.010		0.010	2.000	0.010	7.000	0.010	0.010	1.000
2	2.000	0.010	3.000	0.010		0.050	2.250	0.050	7.250	0.050	0.050	1.250
3	2.000	0.010	3.000	0.010		0.100	2.500	0.100	7.500	0.100	0.100	1.500
4	2.000	0.050	3.250	0.050		0.050	2.500	0.050	7.500	0.050	0.100	1.250
5	2.000	0.050	3.250	0.050	.	0.100	2.000	0.100	7.000	0.100	0.010	1.500
6	2.000	0.050	3.250	0.050	.	0.010	2.250	0.010	7.250	0.010	0.050	1.000
7	2.000	0.100	3.500	0.100	.	0.100	2.250	0.100	7.250	0.100	0.050	1.500
8	2.000	0.100	3.500	0.100		0.010	2.500	0.010	7.500	0.010	0.100	1.000
9	2.000	0.100	3.500	0.100		0.050	2.000	0.050	7.000	0.050	0.010	1.250
10	2.250	0.010	3.000	0.050		0.050	2.500	0.050	7.500	0.050	0.100	1.250
...				
44	2.250	0.100	3.000	0.050		0.100	2.000	0.010	7.500	0.100	0.010	1.250
45	2.250	0.100	3.000	0.050		0.010	2.250	0.050	7.000	0.010	0.050	1.500
46	2.500	0.010	3.500	0.050		0.050	2.500	0.100	7.250	0.010	0.010	1.250
47	2.500	0.010	3.500	0.050		0.100	2.000	0.010	7.500	0.050	0.050	1.500
48	2.500	0.010	3.500	0.050	.	0.010	2.250	0.050	7.000	0.100	0.100	1.000
49	2.500	0.050	3.000	0.100	.	0.100	2.250	0.010	7.000	0.050	0.100	1.500
50	2.500	0.050	3.000	0.100	.	0.010	2.500	0.050	7.250	0.100	0.010	1.000
51	2.500	0.050	3.000	0.100		0.050	2.000	0.100	7.500	0.010	0.050	1.250
52	2.500	0.100	3.250	0.010		0.010	2.000	0.050	7.500	0.100	0.050	1.000
53	2.500	0.100	3.250	0.010		0.050	2.250	0.100	7.000	0.010	0.100	1.250
54	2.500	0.100	3.250	0.010	...	0.100	2.500	0.010	7.250	0.050	0.010	1.500

**Figure 3.** Run Times of Simulated Annealing (SA) and Exhaustive Search (ES) Algorithms.

As can be observed from Figure 3, the exhaustive search required significantly more time to find the optimum solution in each experiment. Both the SA and exhaustive search models were able to obtain the optimum solution. Additionally, a comparison is conducted with all disassembly sequences generated in each of the 1000 trials to validate the results.

To prove the effectiveness of simulated annealing (SA) over exhaustive search (ES) in the proposed experiments, a hypothesis test for mean can be utilized to statistically demonstrate the superiority of SA's performance over the performance of ES.

The test is initiated with $n = 54$ with a significance level, $\alpha = 0.05$. Here, μ_{SA} represents the runtime mean of the SA experiments, and μ_{ES} represents the mean execution time of the ES experiments. The hypothesis is as follows:

$$H_0 : \mu_{SA} \geq \mu_{ES}$$

$$H_1 : \mu_{SA} < \mu_{ES}$$

From results represented in Figure 3, $\mu_{SA} = 1128$ s with standard deviation $S_{SA} = 152$ s. For the ES, $\mu_{ES} = 2223$ s with a significantly low standard deviation, hence, is neglected. Based on these values, the t-value can be calculated as in Equation (1):

$$t = \frac{\mu_{SA} - \mu_{ES}}{\frac{S_{SA}}{\sqrt{n}}} = -52.94 \quad (1)$$

The critical-value of the t-distribution with a degree of freedom $df = 53$ and $\alpha = 0.05$, is (-1.67) . Therefore, the null hypothesis (H_0) is rejected and the assumption of $\mu_{SA} < \mu_{ES}$ is accepted to be correct.

4. Disassembly Line Balancing Using Task Allocation

Despite the fact that a single robot arm or a disassembly cell provides more flexibility, disassembly lines rise as a higher efficiency alternative for automated disassembly, and hence, are the preferred alternative for such settings [39]. Therefore, building on [39], this study proposes a multi-resource automated disassembly line with balanced task allocation. Similar to disassembly sequence generation, disassembly line balancing problems are also proven to be NP-complete, justifying the utilization of metaheuristic algorithms. It is worth noting that the problem is solved with the optimum number of robot arms as proposed by NP-complete [39], and also for the constant number of robot arms to be compatible with the real life scenarios. Utilizing Simulated Annealing (SA), the following describes a three-robot-arm disassembly line. For more information on disassembly line balancing, see [39–47].

The introduction of multiple robot arms to the problem environment, ensuring that all robot arms work with a balanced load, is achieved via Equation (2).

$$x_{jk} = \begin{cases} 1 & \text{if part } j \text{ is assigned to station } k \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$[x_{jk}]_{n \times m}, (n) \text{ number of parts, and } (m) \text{ number of machines} \quad (3)$$

where, x_{jk} represents the time required to disassemble the component and the total load on the current robot arm. In Equation (3), the variable n represents the number of discovered items, and m represents the number of available robot arms for disassembly.

$$c = \left(\sum_{i=1}^n dt_i \right) / n \quad (4)$$

In Equation (4), the variable c represents cycle time, viz., and maximum time available at each workstation, whereas dt is the disassembly time for all available items (i). This equation always sets the value of c to the average of disassembly time dt . The value of c becomes part of the evaluation function.

The first evaluation function is represented in Equation (5). The main factor in this evaluation function is to minimize the number of robots running while keeping the entire system balanced. The number of robot arms is set to a constant value of 5.

$$\min f_1 = m \quad (5)$$

In Equation (6), main factor is balance the load on all the robot arms, and this is applied by calculating the square difference between the constant factor from Equation (5) and the total time the robot arm is running.

$$\min f_2 = \sum_{j=1}^m (c - ST_j)^2 \quad (6)$$

Disassembling the hazardous items has priority over other components to ensure the environmentally-benign nature of the algorithm. This condition can be represented as follows:

$$\min f_3 = \sum_{i=1}^n i \times h_{PS_i}, \quad h_{PS_i} = \begin{cases} 1 & \text{hazardous} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

The final evaluation function is represented in Equation (8). This equation represents the demand, and is based on positive integer values that indicate the quantity required of a given part after it is removed (or 0 if it is not desired) and its position in the sequence:

$$\min f_4 = \sum_{i=1}^n i \times d_{PS_i}, \quad d_{PS_i} \in N, PS_i \quad (8)$$

subject to

$$\sum_{k=1}^m X_{ji} = 1, \quad j = 1, \dots, n \quad (9)$$

Figure 4 represents the combined SA disassembly sequence generation and SA task allocation. The parameters used for the SA algorithm is identical to the parameters utilized in [4], where the exponential annealing parameter, $\alpha = 0.99$, the initial temperature, $T_0 = 500$, with a proportional new temperature function. In the initial step, the system detects available items for disassembly and, if found, the SA algorithm is run to generate the disassembly sequence for the list of items. The optimum or near-optimum solution will be passed to the third phase to allocate tasks and find the optimum task allocation using SA. The process will continue until all the items are disassembled successfully. The results are presented in Table 4.

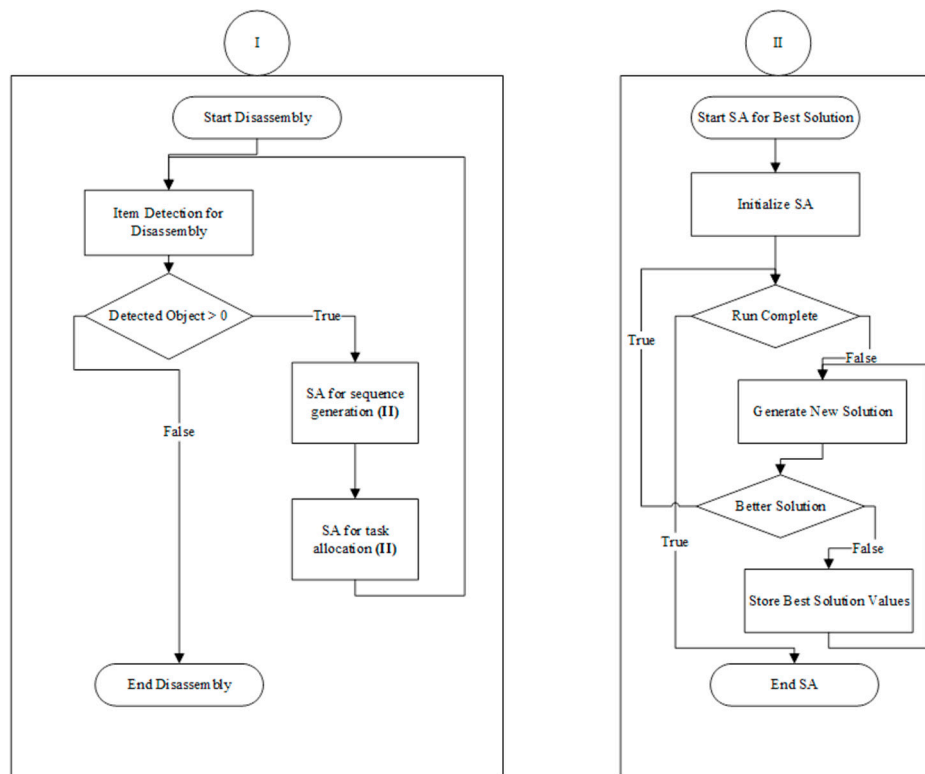


Figure 4. Disassembly sequence and task allocation process.

Table 4. Disassembly sequence and task allocation results.

Iteration	Sequence	Allocation	Fitness Value
1	0 1 D r A 24.7381	[1 0 0]	1, 3, 0, 0
		[3 0 0]	
2	0 2 3 4 DNN ruu CPP 49.1646	[1 0 0]	3, 6, 0, 0
		[0 1 0]	
		[0 0 1]	
		[6 0 0]	
		[0 3 0]	
3	0 7 8 6 5 NNNN rrru AAAP 73.8853	[0 0 1]	3, 5, 0, 0
		[0 1 0]	
		[5 0 0]	
		[0 3 0]	
		[0 0 4]	
		[0 4 0]	
		[1 0 0]	
		[0 1 0]	
4	0 10 9 DN rs AP 53.2419	[3 0 0]	2, 6, 0, 0
		[0 2 0]	

As can be observed from Table 4, task allocation allows multiple robots to disassemble the product components while sustaining load balancing on robot arms.

5. Conclusions and Discussion

This study, built on an environmentally-benign and economically-feasible disassembly sequencing approach, proposes an automated multi-resources disassembly platform. In order to evaluate the performance of the previously-published SA model, the algorithm has been rigorously tested by comparing its performance to exhaustive search. A robustness analysis using an orthogonal array design has been conducted to demonstrate the flexibility of the algorithm. To achieve this, several experiments have been designed with all parameters altered simultaneously, each parameter covering a wide range of values. Reducing the number of experiments from 324 to 54, an L54 ($2^1 \times 3^{25}$) orthogonal array has been employed to design the experiments as opposed to a full factorial design. The results indicate that the proposed model, in addition to its ability to produce (near) optimal solution(s) with efficiency, is also robust, producing reliable results under varying conditions of the disassembly platform. The multi-criteria oriented algorithm is applicable to a large variety of product structures while also eliminating the need for the availability of BOM data for landfilled EOL products. Compared to its counterparts, the algorithm has the ability to incorporate state-of-the art technologies such as image recognition and automated multi-arm robotic disassembly platforms. Following this, the SA model is improved to incorporate multiple robot arms. A task sequencing algorithm has been utilized to create a balanced disassembly line. Combined with the work described in [4], the resulting approach is able to (i) incorporate the preferences of decision makers, (ii) remain loyal to the EOL product structure by preserving the relationships, (iii) eliminate the uncertainty in the overall system by utilizing image recognition, and (iv) provide robust, reliable and efficient solution to the disassembly sequencing problem in a multi-resource environment.

In any optimization problem, there are two major issues that need to be considered: resource utilization and execution time. Serverless architecture detects resources for the problem to be

executed without interference with other processes. The architecture provides a complete tool, allowing its users to monitor the processing status via a manageable dashboard. The dashboard takes process-related elements into account, including latency, real-time processing, background processing, batch processing, concurrency, memory limits, processing time limit, and synchronous versus asynchronous processing. However, as also indicated by the results of the literature survey, the serverless architecture is still a relatively new concept, and it has not been considered for automated disassembly systems where multiple robot arms are present. Therefore, it is recommended that related future research concentrate on creating solutions for the disassembly sequencing and disassembly line balancing considering efficient online platforms.

Author Contributions: Conceptualization, A.E., E.K. and S.M.G.; methodology, M.A., A.E., E.K., S.M.G. and T.S.; software, M.A.; validation, A.E. and E.K.; formal analysis, M.A., A.E., E.K., S.M.G. and T.S.; investigation, M.A., A.E., E.K., S.M.G. and T.S.; resources, M.A., A.E., E.K., S.M.G. and T.S.; data curation, M.A. and E.K.; writing—original draft preparation, M.A., A.E., E.K., S.M.G. and T.S.; writing—review and editing, M.A., A.E., E.K., S.M.G. and T.S.; visualization, M.A., A.E., E.K., S.M.G. and T.S.; supervision, E.K., S.M.G. and T.S.; project administration, E.K.; funding acquisition, A.E., E.K., T.S.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Baldé, C.P.; Forti, V.; Gray, V.; Kuehr, R.; Stegmann, P. *The Global E-Waste Monitor 2017 Quantities, Flow, and Resources*; The Global E-waste Monitor—2017, United Nations University (UNU); International Telecommunication Union (ITU); International Solid Waste Association (ISWA): Bonn, Germany; Geneva, Switzerland; Vienna, Austria, 2017.
2. McGovern, S.M.; Gupta, S.M. A balancing method and genetic algorithm for disassembly line balancing. *Eur. J. Oper. Res.* **2007**, *179*, 692–708. [[CrossRef](#)]
3. Kalayci, C.B.; Gupta, S.M. Simulated annealing algorithm for solving sequence-dependent disassembly line balancing problem. *IFAC Proc. Vol.* **2013**, *46*, 93–98. [[CrossRef](#)]
4. Alshibli, M.; El Sayed, A.; Tozanli, O.; Kongar, E.; Sobh, T.M.; Gupta, S.M. A Decision Maker-Centered End-of-Life Product Recovery System for Robot Task Sequencing. *J. Intell. Robot. Syst.* **2017**, *91*, 603–616. [[CrossRef](#)]
5. El Sayed, A.; Kongar, E.; Gupta, S.M. A genetic algorithm approach to end-of-life disassembly sequencing for robotic disassembly. In Proceedings of the Northeast Decision Sciences Institute Conference, NEDSI 2010, Alexandria, VA, USA, 26–28 March 2010.
6. El Sayed, A.; Kongar, E.; Gupta, S.M.; Sobh, T. A Robotic-Driven Disassembly Sequence Generator for End-Of-Life Electronic Products. *J. Intell. Robot. Syst.* **2012**, *68*, 43–52. [[CrossRef](#)]
7. Zhang, C.; Chen, M. Designing and verifying a disassembly line approach to cope with the upsurge of end-of-life vehicles in China. *Waste Manag.* **2018**, *76*, 697–707. [[CrossRef](#)] [[PubMed](#)]
8. Feng, Y.; Gao, Y.; Tian, G.; Li, Z.; Hu, H.; Zheng, H. Flexible Process Planning and End-of-Life Decision-Making for Product Recovery Optimization Based on Hybrid Disassembly. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 311–326. [[CrossRef](#)]
9. Alshibli, M.; El Sayed, A.; Kongar, E.; Sobh, T.M.; Gupta, S.M. Disassembly Sequencing Using Tabu Search. *J. Intell. Robot. Syst.* **2016**, *82*, 69–79. [[CrossRef](#)]
10. Sait, S.M.; Youssef, H. *Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems*; IEEE Computer Society Press: Washington, DC, USA, 1999; p. 387.
11. Alagumurthi, N.; Palaniradja, K.; Soundararajan, V. Optimization of grinding process through design of experiment (DOE)—A comparative study. *Mater. Manuf. Process.* **2006**, *21*, 19–21. [[CrossRef](#)]
12. Kondapalli, S.P.; Chalamalasetti, S.R.; Damara, N.R. Application of Taguchi based design of experiments to fusion arc weld processes: A review. *Int. J. Bus. Res. Dev.* **2013**, *4*. [[CrossRef](#)]
13. Freddi, A.; Salmon, M. Introduction to the Taguchi Method. In *Design Principles and Methodologies: From Conceptualization to First Prototyping with Examples and Case Studies*; Springer International Publishing: Cham, Germany, 2019; pp. 159–180.

14. Ilgin, M.A.; Taşoğlu, G.T. Simultaneous Determination of Disassembly Sequence and Disassembly-to-Order Decisions Using Simulation Optimization. *J. Manuf. Sci. Eng.* **2016**, *138*, 101012. [CrossRef]
15. Tu, C.-S.; Huang, H.-S.; Tsai, M.-T.; Cheng, F.-S. Bee colony optimization with Taguchi method for solving the dynamic economic dispatch. *MATEC Web Conf.* **2018**, *185*, 00033. [CrossRef]
16. Chang, Y.-P. Optimal harmonic filters design of the Taiwan high speed rail traction system of distributor generation system with specially connected transformers. *Int. J. Electr. Power Energy Syst.* **2014**, *62*, 80–89. [CrossRef]
17. Gungor, A.; Gupta, S.M.; Pochampally, K.; Kamarthi, S.V. Complications in disassembly line balancing. *Proc. SPIE* **2001**, 4193. [CrossRef]
18. Mete, S.; Çil, Z.A.; Özceylan, E.; Ağpak, K. Resource constrained disassembly line balancing problem. *IFAC Pap. Online* **2016**, *49*, 921–925. [CrossRef]
19. Bentaha, M.L.; Battaïa, O.; Dolgui, A. Disassembly line balancing and sequencing under uncertainty. *Proc. CIRP* **2014**, *15*, 239–244. [CrossRef]
20. Morgan, S.D.; Gagnon, R.J. A systematic literature review of remanufacturing scheduling. *Int. J. Product. Res.* **2013**, *51*, 4853–4879. [CrossRef]
21. Gungor, A.; Gupta, S.M. A solution approach to the disassembly line balancing problem in the presence of task failures. *Int. J. Product. Res.* **2001**, *39*, 1427–1467. [CrossRef]
22. Torres, F.; Gil, P.; Puente, S.; Pomares, J.; Aracil, R. Automatic PC disassembly for component recovery. *Int. J. Adv. Manuf. Technol.* **2004**, *23*, 39–46. [CrossRef]
23. Gutjahr, A.L.; Nemhauser, G.L. An algorithm for the line balancing problem. *Manag. Sci.* **1964**, *11*, 308–315. [CrossRef]
24. Erel, E.; Gokcen, H. Shortest-route formulation of mixed-model assembly line balancing problem. *Eur. J. Oper. Res.* **1999**, *116*, 194–204. [CrossRef]
25. McMullen, P.R.; Tarasewich, P. Using ant techniques to solve the assembly line balancing problem. *IIE Trans.* **2003**, *35*, 605–617. [CrossRef]
26. Duta, L.; Filip, F.G.; Henrioud, J.-M. Applying equal piles approach to disassembly line balancing problem. *IFAC Proc. Vol.* **2005**, *38*, 152–157. [CrossRef]
27. Duta, L.; Filip, F.G.; Henrioud, J.-M. A method for dealing with multi-objective optimization problem of disassembly processes. In Proceedings of the IEEE International Symposium on Assembly and Task Planning, Besancon, France, 11 July 2003; pp. 163–168.
28. Avikal, S.; Mishra, P.; Jain, R.; Yadav, H. A PROMETHEE method based heuristic for disassembly line balancing problem. *Ind. Eng. Manag. Syst.* **2013**, *12*, 254–263. [CrossRef]
29. McGrath, G.; Brenner, P.R. Serverless computing: Design, implementation, and performance. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), Atlanta, GA, USA, 5–8 June 2017; pp. 405–410.
30. Ho, T.K.; Lee, J.Y. A row-permuted data reorganization algorithm for growing server-less video-on-demand systems. In Proceedings of the CCGrid 2003, 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, Tokyo, Japan, 12–15 May 2003; pp. 44–51.
31. Bolosky, W.J.; Douceur, J.R.; Ely, D.; Theimer, M. Feasibility of a serverless distributed file system deployed on an existing set of desktop PCs. *ACM SIGMETRICS Perform. Eval. Rev.* **2000**, *28*, 34–43. [CrossRef]
32. Hendrickson, S.; Sturdevant, S.; Harter, T.; Venkataramani, V.; Arpaci-Dusseau, A.C.; Arpaci-Dusseau, R.H. Serverless computation with openlambda. *Elastic* **2016**, *60*, 80.
33. Bila, N.; Dettori, P.; Kalso, A.; Watanabe, Y.; Youssef, A. Leveraging the serverless architecture for securing linux containers. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), Atlanta, GA, USA, 5–8 June 2017; pp. 401–404.
34. Baldini, I. The serverless trilemma: Function composition for serverless computing. In Proceedings of the 2017 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, Vancouver, BC, Canada, 25–27 October 2017.
35. Adya, A.; Bolosky, W.J.; Cermak, G.; Douceur, J.R.; Theimer, M.M.; Wattenhofer, R.P. Serverless Distributed File System. U.S. Patent 7062490B2, 2006. Available online: <https://patents.google.com/patent/US7062490B2/en> (accessed on 8 December 2018).
36. Ilgin, M.A.; Gupta, S.M. *Multiple Criteria Decision Making Applications in Environmentally Conscious Manufacturing and Product Recovery*; CRC Press: Boca Raton, FL, USA, 2017.

37. Taguchi, G.; Wu, Y. *Introduction to Off-Line Quality Control*; Central Japan Quality Control Association: Nagoya, Japan, 1979.
38. Phadke, M.S. *Quality Engineering Using Robust Design*; Prentice Hall PTR: Upper Saddle River, NJ, USA, 1995.
39. Kalayci, C.B.; Gupta, S.M.; Nakashima, K. A simulated annealing algorithm for balancing a disassembly line. In *Design for Innovative Value towards a Sustainable Society*; Springer: Berlin, Germany, 2012; pp. 714–719.
40. Kalayci, C.B.; Gupta, S.M. Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem. *Expert Syst. Appl.* **2013**, *40*, 7231–7241. [[CrossRef](#)]
41. Kalayci, C.B.; Gupta, S.M. A particle swarm optimization algorithm with neighborhood-based mutation for sequence-dependent disassembly line balancing problem. *Int. J. Adv. Manuf. Technol.* **2013**, *69*, 197–209. [[CrossRef](#)]
42. Kalayci, C.B.; Gupta, S.M. Ant colony optimization for sequence-dependent disassembly line balancing problem. *J. Manuf. Technol. Manag.* **2013**, *24*, 413–427. [[CrossRef](#)]
43. Kalayci, C.B.; Polat, O.; Gupta, S.M. A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem. *Ann. Oper. Res.* **2016**, *242*, 321–354. [[CrossRef](#)]
44. Kalayci, C.B.; Gupta, S.M. A tabu search algorithm for balancing a sequence-dependent disassembly line. *Product. Plan. Control* **2014**, *25*, 149–160. [[CrossRef](#)]
45. Kalayci, C.B.; Gupta, S.M. Tabu search for disassembly line balancing with multiple objectives. In Proceedings of the 41st International Conference on Computers and Industrial Engineering, Los Angeles, CA, USA, 23–26 October 2011.
46. Kalayci, C.B.; Hancilar, A.; Gungor, A.; Gupta, S.M. Multi-objective fuzzy disassembly line balancing using a hybrid discrete artificial bee colony algorithm. *J. Manuf. Syst.* **2015**, *37*, 672–682. [[CrossRef](#)]
47. Kalayci, C.B.; Gupta, S.M. River formation dynamics approach for sequence-dependent disassembly line balancing problem. In *Reverse Supply Chains: Issues and Analysis*; CRC Press: Boca Raton, FL, USA, 2013; pp. 289–312.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).