COGNITIVE ROBOTIC DISASSEMBLY SEQUENCING FOR ELECTROMECHANICAL END-OF-LIFE PRODUCTS VIA DECISION-MAKER-CENTERED HEURISTIC OPTIMIZATION ALGORITHM

Mohammad Alshibli

Under the Supervision of Dr. Elif Kongar

DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

AND ENGINEERING

THE SCHOOL OF ENGINEERING

UNIVERSITY OF BRIDGEPORT

CONNECTICUT

May, 2018

COGNITIVE ROBOTIC DISASSEMBLY SEQUENCING FOR ELECTROMECHANICAL END-OF-LIFE PRODUCTS VIA DECISION-MAKER-CENTERED HEURISTIC OPTIMIZATION ALGORITHM

Mohammad Alshibli

Under the Supervision of Dr. Elif Kongar

Approvals

Committee Members

Name	Signature	Date
Dr. Elif Kongar	Elif Kongar	4/26/18
Dr. Tarek M. Sobh	T. Sobh	4/26/2018
Dr. Christian Bach		4/26/2018
Dr. Ausif Mahmood	Ausif Mahmood.	4-26-2018
Dr. Surendra M. Gupta	S. M. Gupta	4/26/2018

Ph.D. Program Coordinator

Dr. Khaled M. Elleithy		4/26/18

Chairman, Computer Science and Engineering Department

Dr. Ausif Mahmood	Ausif Mahmood.	4-26-2018

Dean, School of Engineering

Dr. Tarek M. Sobh	T. Sobh	4/26/2018

COGNITIVE ROBOTIC DISASSEMBLY SEQUENCING FOR ELECTROMECHANICAL END-OF-LIFE PRODUCTS VIA DECISION-MAKER-CENTERED HEURISTIC OPTIMIZATION ALGORITHMS

ABSTRACT

End-of-life (EOL) disassembly has developed into a major research area within the sustainability paradigm, resulting in the emergence of several algorithms and models to solve related problems. End-of-life disassembly focuses on regaining the value added into products which are considered to have completed their useful lives due to a variety of reasons such as lack of technical functionality and/or lack of demand. Disassembly is known to possess unique characteristics due to possible changes in the EOL product structure and hence, cannot be considered as the reverse of assembly operations. With the same logic, obtaining a near-optimal/optimal disassembly sequence requires intelligent decision making during the disassembly when the sequence need to be regenerated to accommodate these unforeseeable changes. That is, if one or more components which were included in the original bill-of-material (BOM) of the product is missing and/or if one or more joint types are different than the ones that are listed in the original BOM, the sequencer needs to be able to adapt and generate a new and accurate alternative for disassembly. These considerations require disassembly sequencing to be solved by highly adaptive methodologies justifying the utilization of image detection technologies for online real-time disassembly. These methodologies should also be capable of handling efficient search techniques which would provide equally reliable but faster solutions compared to their exhaustive search counterparts. Therefore, EOL disassembly sequencing literature offers a variety of heuristics techniques such as Genetic Algorithm (GA), Tabu Search (TS), Ant Colony Optimization (ACO), Simulated Annealing (SA)

and Neural Networks (NN). As with any data driven technique, the performance of the proposed methodologies is heavily reliant on the accuracy and the flexibility of the algorithms and their abilities to accommodate several special considerations such as preserving the precedence relationships during disassembly while obtaining near-optimal or optimal solutions. This research proposes three approaches to the EOL disassembly sequencing problem. The first approach builds on previous disassembly sequencing research and proposes a Tabu Search based methodology to solve the problem. The objectives of this 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 results of the previous Genetic Algorithm with the proposed Tabu Search Algorithm is also included. Following this, the disassembly sequencing problem is further investigated by introducing an automated disassembly framework for end-of-life electronic products. This proposed model is able to incorporate decision makers' (DMs') preferences into the problem environment for efficient material and component recovery. The proposed disassembly sequencing approach is composed of two steps. The first step involves the detection of objects and deals with the identification of precedence relationships among components. This stage utilizes the BOMs of the EOL products as the primary data source. The second step identifies the most appropriate disassembly operation alternative for each component. This is often a challenging task requiring expert opinion since the

decision is based on several factors such as the purpose of disassembly, the disassembly method to be used, and the component availability in the product. Given that there are several factors to be considered, the problem is modeled using a multi-criteria decision making (MCDM) method. In this regard, an Analytic Hierarchy Process (AHP) model is created to incorporate DMs' verbal expressions into the decision problem while validating the consistency of findings. These results are then fed into a metaheuristic algorithm to obtain the optimum or near-optimum disassembly sequence. In this step, a metaheuristic technique, Simulated Annealing (SA) algorithm, is used.

In order to test the robustness of the proposed Simulated Annealing algorithm an experiment is designed using an Orthogonal Array (OA) and a comparison with an exhaustive search is conducted. In addition to testing the robustness of SA, a third approach is simultaneously proposed to include multiple stations using task allocation. Task allocation is utilized to find the optimum or near-optimum solution to distribute the tasks over all the available stations using SA. The research concludes with proposing a serverless architecture to solve the resource allocation problem. The architecture also supports non-conventional solutions and machine learning which aligns with the problems investigated in this research. Numerical examples are provided to demonstrate the functionality of the proposed approaches.

ACKNOWLEDGEMENTS

My thanks to God who has helped me complete this work successfully. I am also grateful to my family for their understanding and continuous encouragement.

I would like to express my special appreciation and thanks to my advisors, Dr. Elif Kongar and Dr. Tarek M. Sobh for their tremendous mentorship. Thank you for encouraging my research and for allowing me to grow as a research scientist. Your advice on both my research as well as on my career have been priceless.

I also would like to thank to my committee members, Dr. Surendra M. Gupta, Dr. Ausif Mahmood and Dr. Christian Bach, for their constant and constructive feedback. My dissertation work has significantly improved thanks to their valuable contributions.

I would like to dedicate this work to the souls of my father and my mother. With this dissertation their dream came true. I would like to express my gratitude to my family, my siblings, my wife and my children. Thank you for all your time and support during this long journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Research Problem and Scope

Decision making is a scientific approach for solving problems [4, 5] and is considered to be a crucial step for many organizations in solving short- and long-range problems. Since the acceptance of its vital role, several decision-making methods have been developed to map complex operations and to incorporate various factors into the modeling environment such as uncertainty, preferences of decision makers and the expected value of decision alternatives. Decision making is generally investigated under Operations Research (OR); a technique that summarizes the major elements of the problem, integrating them into a mathematical model to solve and analyze the model to obtain the optimum or near-optimum solution [6]. Ackoff and Sasieni [7] describes operations research as the application of scientific method, by interdisciplinary teams, to problems involving the control of organized systems to provide efficient solutions which are compatible with goals of companies. Operations research is also known to improve the efficiency and the effectiveness of an organization as it is capable of improving quality, reducing costs and minimizing risks [8].

According to Taha [9], in order to be able to solve decision making problems, the decision alternatives need to be mapped, the restrictions on the model environment need to be included, and the objective function need to be created to evaluate the alternatives.

Lieberman brings a similar perspective and defines operations research as a tool to be applied to problems that are concerned with how to conduct and coordinate the operations within an organization [10]. Generally, achieving this understanding relies on combining analytical and numerical techniques.

A typical OR project consists of three steps: (1) Model building, (2) Model solution, and (3) Implementation and analysis of findings. The emphasis of this research is on the second step which involves scientific methodologies or techniques. These techniques are analytical in nature and can be categorized in one of four categories; simulation techniques, mathematical analysis techniques, optimum-seeking techniques, and heuristics.

Metaheuristics such as genetic algorithms, Tabu Search, evolutionary programming, and Simulated Annealing are relatively new heuristics algorithms. These methods are considered to be more simplistic and effective compared to their counterparts. Sorensen and Glover defines a metaheuristic as a high-level problem independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms [11]. Metaheuristics attempts to combine exact algorithms with heuristics.

Using metaheuristics, this research primarily focuses on regaining the value embedded in the manufactured products, in particular, electronic products which have completed their useful lives. Due to increasing volumes of e-waste and scarcity of raw materials, the utilization and subsequent re-utilization of recyclable materials and reusable components are often cited as the most viable solutions to reducing user waste.

End-of-life (EOL) disassembly is the subsequent step following the collection of EOL products to regain the value embedded in e-waste. The EOL products are defined as the products which completed their usefulness due to deterioration and/or obsolescence. The need for disassembly originates from economic, social and environmental concerns since disassembly of EOL products plays an important role in making available part or materials for reuse, reducing the amount of industrial waste and decreasing environmental deterioration.

Further, the factor of 'uncertainty' is considered as a unique characteristic of disassembly. The implied uncertainty stems from the probable changes on the product during its life cycle or from the likely damage that occurs after the product is landfilled. Therefore, disassembly cannot be considered as the reverse of assembly. Obtaining the optimum or near-optimum disassembly sequence is a complex problem, thus using a conventional, exhaustive search is generally considered to be mathematically prohibitive. Therefore, solving the disassembly sequence problem using metaheuristic approaches rises as a more effective way to find the optimum or near-optimum solution. Furthermore, solving the disassembly sequencing problem with operations research techniques requires the model to be flexible enough to accommodate unexpected changes in the model environment, making metaheuristics better alternatives for this purpose.

1.2 Research Motivation

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. Furthermore, 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 reuse, recycling, or proper disposal. Within responsive products, electrical and electronic equipment is the largest growing waste stream requiring economically and environmentally solid and efficient reverse logistics and supply chain operations. Waste electrical and electronic equipment (WEEE) uses large quantities of natural resources, including substantial amounts of precious metals such as gold, silver, and copper during their production. Furthermore, WEEE 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 reuse activities economically valid. Reuse, 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. Since using human labor to disassemble these products significantly increases the overall cost and time of the recovery system, the need for utilizing automated solutions gains importance. 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 repair components that are not accessible by humans, making robotic solutions the only viable alternative.

The problem of generating an optimal sequence for disassembly operations is rather challenging due to the uncertainty of the process. Electrical and electronic equipment are often subject to various changes in their original bill-of-materials due to technological compatibility issues. 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. This fact often mandates destructive disassembly, prohibiting the reuse of still functioning components.

With these motivations, this work initially aims to decrease the uncertainty in disassembly processes and to address the aforementioned challenges by introducing two modules: A sensory system, and an online Tabu Search algorithm [12].

The sensory system is used for identifying the depth of the product with the help of a digital camera that captures product images for processing and detecting the components. The algorithm then generates an online real-time disassembly sequence using this information, hence overcoming the uncertainty in the product structure. Building on this model, the proposed disassembly system has been improved to accommodate the preferences of the decision makers, a more efficient metaheuristics algorithm and multiple robot arms. Performance comparisons over exhaustive searches are provided and the robustness of the proposed model is tested and validated using orthogonal arrays.

The following provides detailed information regarding the motivation and potential impact of this work on the environment, economy and the society in general.

Due to shortening life cycles of electronic products and increasing need for faster and more reliable technologies, the demand for raw materials in related industries is increasing to meet the production-line requirements. Raw materials and components used in technological products are often limited and valuable in nature. For instance, computer production uses gold, copper, tin, silver and several other precious metals. Since the demand for state-of-the-art technology products is growing along with the technological advances, finding alternative sources to fulfill the production line requirements is important. Shortening life cycles of technological products have also led to substantial amounts of electronic waste (e-waste). According to the U.S. Environmental Protection Agency (EPA) [13], on average, approximately 416 thousand mobile phones and 142 thousand computers are discarded daily. As a result of discarded consumer electronics, annual e-waste has reached more than 3 million tons over the past decade.

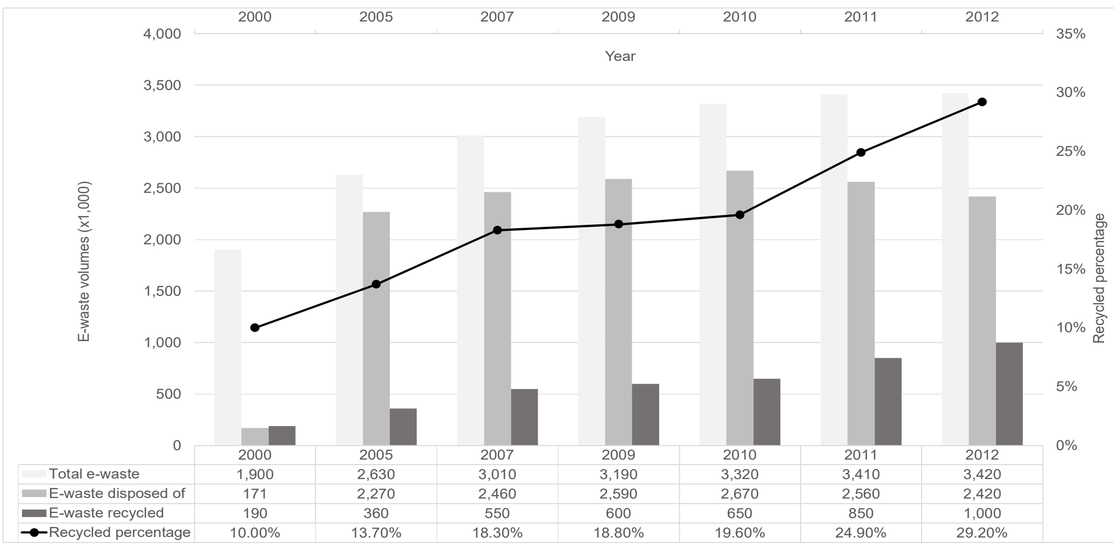
Consequently, environmental awareness has increased worldwide, with governments and related agencies enforcing rules and regulations encouraging industries to expand their environmentally-benign operations. One way to address this issue is to restructure the product life cycle to regain the value added into the electronic waste.

End-of-life electronics recovery is proven to be economically viable when conducted appropriately, in addition to its positive impact on the environment and society as a whole. For instance, it is reported that for every 1 million recycled mobile devices, 20 thousand lbs. of copper, 550 lbs. of silver, 50 lbs. of gold and 20 lbs. of palladium can be recovered [14]. To further strengthen the argument that the electronic waste recovery operations are effective in regaining the value added to EOL products, Apple reported having recovered over 61 million pounds of materials (Table 1.1) from returned retail products [15].

Material	**Quantity (lb.)**
Steel	23,101,000
Plastics	13,422,360
Glass	11,945,680
Aluminum	4,518,200
Copper	2,953,360
Cobalt	189,544
Zinc	130,036
Lead	44,080
Nickel	39,672
Silver	6,612
Tin	4,408
Gold	2,204

Table 1.1 Amount of material recovered through take-back initiatives in 2015 [15].

Figure 1.1 illustrates the total e-waste generated in the United States and the amounts of disposal and recycling between 2000 and 2012. As illustrated in the figure, the total amount of e-waste has increased annually over the same period along with the percentage of recycled e-waste. One plausible explanation for this trend is growing economically-viable and environmentally-sustainable practices, fostered by increasing environmental awareness. Despite the growing efforts and the significant potential gain, however, there are still large volumes of precious materials which are not recovered and ultimately landfilled, requiring economically- and environmentally-benign EOL recovery solutions.



	2000	2005	2007	2009	2010	2011	2012
Total e-waste	1,900	2,630	3,010	3,190	3,320	3,410	3,420
E-waste disposed of	171	2,270	2,460	2,590	2,670	2,560	2,420
E-waste recycled	190	360	550	600	650	850	1,000
Recycled percentage	10.00%	13.70%	18.30%	18.80%	19.60%	24.90%	29.20%

Figure 1.1 Total e-waste generation, disposal and recycling in the United States [13].

These attempts to regain the value added into the EOL electronic products are forcing businesses to consider several additional cost factors such as the costs associated with EOL product take back, collection, disassembly and recovery operations. Given the environmental nature of these operations, coupled with their potential societal impact, the problem can be appropriately addressed only via a carefully-designed EOL management system that considers these three factors simultaneously. EOL management systems comprise of multiple consecutive steps. The first step 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 the disassembly of these EOL products. Disassembly aims at regaining the value added into the EOL products via recycling or reuse, or alternatively, storing the components for possible future use or properly disposing of them to minimize the environmental hazard. Furthermore, according to the Global e-waste monitor, 44.7 million metric tons of e-waste were generated globally in 2016 (Figure 1.2), with an estimated e-waste per individual around 6.1 kg [16].

Today, several industries adopted varying levels of take back policies and are attempting to disassemble their own products for their material and component content. These companies rely heavily on the original blueprints of these products when balancing their disassembly lines with the assumption that the original product structure remained unaltered during use. However, EOL products are known to be associated with high levels of uncertainty due to the changes which are likely to occur in the product structure during their useful lives. There are two reasons why a product reaches its end-of-life; namely, deterioration and obsolescence [17]. Regardless of the reason, components in electronic EOL products are likely to be replaced by other components for upgrade and/or repair purposes. In some cases, product recovery operations are not conducted by the original

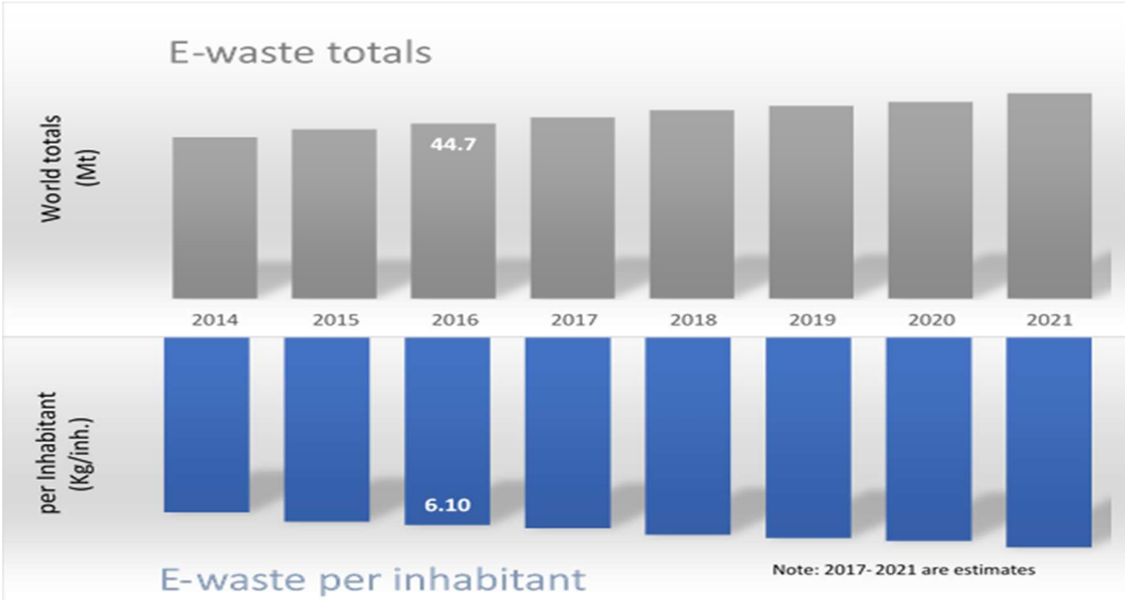


Figure 1.2 E-waste total for years 2014-2021 [16].

equipment manufacturer (OEM), and this task is outsourced to a third-party service provider, resulting in limited access to the original BOMs. Therefore, to ensure the success of disassembly operations in a product recovery chain: an efficient, intelligent, and automated decision-making system is needed. With this motivation, this study proposes environmentally-benign and economically-feasible disassembly sequencing approaches that incorporates decision makers' (DMs') preferences into the modeling environment.

Disassembly sequencing is considered to be an NP-complete problem [18, 19]. As with all NP-complete problems, the complexity of the disassembly sequencing problem increases exponentially with the number of components in the product structure justifying the utilization of metaheuristic methods.

In addition to its NP-complete nature, disassembly sequencing of EOL products requires methods which are able to handle the uncertainty in the product. The problem becomes even more complex since the economic justification of disassembly operations requires large volumes of EOL products with varying levels of demand for their components and materials. That is, the decision maker who is responsible for recovery operations must take into account various factors such as the inventory on hand, the demand for materials and components, and the current market prices prior to disassembly. To address these considerations, this research also presents an Analytic Hierarchy Process (AHP) and Simulated Annealing (SA) - based methodology to generate the optimum or near-optimum disassembly sequence based on the preferences of the decision maker.

1.3 Contributions

This research builds on a previously proposed Genetic Algorithm model for disassembly sequencing and proposes a more efficient 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, 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.

Following this, the research also presents an automated disassembly framework for end-of-life (EOL) electronic products. Proposed model is able to incorporate decision makers' (DMs') preferences into the problem environment for efficient material and component recovery. The disassembly sequencing approach is composed of two steps. The first step involves the detection of objects and deals with the identification of precedence relationships among components. This stage uses the Bill of Materials (BOMs) of the EOL products as the primary data source. The second step identifies the most appropriate disassembly operation alternative for each component. This is often a challenging task requiring expert opinion since the decision is based on several factors, such as the purpose of disassembly, the disassembly method to be used, and the component availability in the product. Given that there are several factors to be considered, the problem is modeled using a multi-criteria decision making (MCDM) method. In this regard, an Analytic Hierarchy Process (AHP) model is created to incorporate DMs' verbal expressions into the decision problem while validating the consistency of findings. These results are then fed into a metaheuristic algorithm to obtain the optimum or near-optimum disassembly sequence. In this step, a metaheuristic technique, Simulated Annealing (SA) algorithm, is used. A numerical example is provided to demonstrate the functionality of the proposed approach.

Building on the results of this model, a robustness test and performance evaluation for the proposed SA disassembly sequence problem are conducted using orthogonal arrays (OAs). The problem is also expanded to have multiple symmetric robot arms as opposed to a single arm. At this stage, all available robot arms (stations) are being deployed and used ensuring the efficiency of their utilization. This solution also utilizes SA to find the optimum task allocation among available stations.

An additional improvement involves the resource allocation. 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, this work recommends an architecture that would help the algorithm acquire required resources to generate the solution in a more efficient manner.

CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

2.1 Tabu Search (TS), Genetic Algorithm (GA), Analytic Hierarchy Process (AHP) and Simulated Annealing (SA) Literature Survey

Evolutionary algorithms have been recognized to be well-suited to multi-objective optimization since their development [20]. 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 [21]. The following summarizes the Tabu Search (TS) and Genetic Algorithm (GA) - related studies in the environmentally conscious manufacturing and robotics fields.

Kongar and Gupta [22] considered the case of complete disassembly utilizing both destructive and non-destructive methods. Their method helped in finding the optimum disassembly sequence faster based on the information from the design process. Therefore, the authors claimed that the algorithm could be used in new product design as well as for recycling and product maintenance. One example of the code for Tabu Search appears in Rizk and ElSayed [23].

McGovern and Gupta [18] 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. [24] 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 was 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.

Torres et al. [25] 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 provided information regarding the type of products and the main components of the product architecture.

ElSayed et al. [26] proposed an online Genetic Algorithm (GA) that aims at handling uncertainty in the EOL product structure. The algorithm consisted 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 $2^1/_2$ D images. The proposed algorithm was able to obtain the optimal disassembly sequence while reducing the time required for disassembling the product.

Xing et al. [27] conducted a survey reviewing 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.

Kalayci and Gupta [28] introduced a Tabu Search (TS) algorithm to solve the Disassembly Line Balancing Problem (DLBP) with multiple objectives. The DLBP described in the paper consisted of multiple objectives requiring the assignment of disassembly tasks to a set of ordered disassembly workstations. The algorithm also satisfied the disassembly precedence constraints and optimized the effectiveness of several measures. The authors assigned the removal of hazardous and high demand components maximum priority.

Torres et al. [29] proposed two types of cooperation among robot arms aiming to manage the task between multiple robots: in the first cooperation, two or more robots cooperated to achieve the same task. In the latter, several tasks were achieved by different robots concurrently. The entire design was built based on a decision tree. The main goal in their follow up work [30] was to retrieve materials from the EOL product via destructive disassembly.

Kuren [31] proposed a disassembly cell prototype and presented a case study for mobile phone disassembly. Since a destructive method was used in this research, the need to preserve precedence relationships has been eliminated in the proposed solution.

This work builds on the algorithms provided in Kongar and Gupta [22]. The proposed genetic algorithm includes Precedence Preservation Crossover (PPX) to accurately reflect 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.

This section first provides an overview of the previous research on Analytic Hierarchy Process (AHP). This is followed by the Simulated Annealing (SA) literature that focused on EOL product recovery systems.

The Analytic Hierarchy Process was first proposed by Prof. Thomas L. Saaty [32] in the early 1970s. AHP is interposed between operational research and decision analysis and is considered as a Multi-Criteria Decision Making (MCDM) method, based on the relative measurement theory. AHP, using linguistic expressions, derives the ratio scales from pairwise comparisons and is designed to help decision makers to make a choice among a set of alternatives [33]. The problem description in AHP consists of goal, criteria, and alternatives.

The AHP process includes the following steps: (i) Decompose problem hierarchy into goal, set of alternatives, and set of criteria, (ii) create a pairwise comparison matrix, (iii) calculate the priority vector for each criterion, and (iv) evaluate the solution by calculating the Consistency Index (CI) and Consistency Ratio (CR) [3, 34]. The final step provides a clear insight into the reliability of the solution. For the priority vector calculations, the literature offers a large variety of methods as also stated by Choo and Wedley [35]. Out of these, eigenvector and eigenvalue, and geometric mean methods are most commonly used. Comparing these, Saaty and Vargas [36] reported eigenvector and eigenvalue based methods to be superior to the methods based on the geometric mean.

The steps required for constructing the main parts of an AHP model are provided by Saaty [37] and include defining: (1) the objective of the decision, (2) the criteria to be selected upon, and (3) the alternatives that may achieve the criteria to reach the goal of the objective. This process is then translated into a mathematical operation depending on the judgments made to assign each criterion a priority when compared with one another.

AHP has been used in several areas such as city evaluation and planning, country ranking, mobile valued service, organ transplant, chess prediction, and facility location [33, 38]. Triantaphyllou and Mann [39] have utilized AHP for a computer upgrade system at a computer integrated manufacturing facility. Similarly, Chakraborty et al. [40] employed AHP to solve a vendor selection problem. The authors determined AHP as the most effective MCDM method due to its ability to provide a near-optimal solution and its capability of handling quantifiable and unquantifiable criteria.

Syamsuddin and Hwang [41] applied AHP to aid decision makers in their efforts to ensure efficient management of information security policies. Dalalah et al. [42] presented a systematic methodology for crane selection. Al-Harbi [43] introduced the application of the AHP method as a potential decision making tool in project management. Koç and Burhan [44] conducted a study to select a location for a new auto glass store using tangible and intangible criteria. Parameshwaran et al. [45] proposed an integrated approach for selecting the most appropriate robot, taking into account both objective and subjective criteria. Wang et al. [46] used the combination of AHP and geographical information system (GIS) to analyze and assess the safety of the shipping routes of the South China Sea.

In another relevant study, Malik et al. [47], studied the characterization and modeling of reverse logistics and claimed that its application was becoming imperative as the issues related to environment and societies gain more and more attention and importance.

In sum, previous literature indicates that the AHP is a viable method when evaluating, comparing and choosing from multiple alternatives. The method also decreases the bias or prejudice in the decision-making process and is considered to be more flexible when compared with other multi-criteria decision making approaches.

Given that hybrid methods are often superior to their standalone counterparts, this research integrated AHP with another well-known heuristics algorithm, Simulated Annealing (SA). SA has been proven to (i) be able to deal with arbitrary systems, (ii) be relatively easy to build, and, more importantly, (iii) be able to produce a faster convergence to the optimal or near-optimal solutions. A performance comparison between SA and Tabu Search was conducted to solve the corridor allocation problem where SA was found to be superior while providing more reliable solutions [48].

Disassembly sequencing literature embodies several heuristics-based methodologies. In one of the most relevant works, Kalayci and Gupta [19] applied simulated annealing to solve the sequence-dependent disassembly line balancing problem (SDDLBP). Following this, the authors applied a variant of the particle swarm optimization algorithm [49], and a hybrid genetic algorithm [50] to the SDDLBP. Kalayci et al. [51] also proposed a hybrid algorithm that combined genetic algorithm with a variable neighborhood search method (VNSGA) to solve the SDDLBP. Table 2.1 provides further details on the methods utilized in related literature including the primary goals and motivations of these studies along with the evaluation criteria and the utilized techniques.

Author(s)	Goal	Motivation	Evaluation Technique(s)	Evaluation Criteria
Kongar and Gupta [52]	Help selecting the desirable disassembly process satisfying several environmental, financial and physical goals.	Environmental	Preemptive integer goal programming	Recycling revenue, total disposal cost, total inventory cost, profit from resale, no. of items stored, no. of recycled items, and no. of disposed items.
Massoud and Gupta [53]	Determining the best combination of EOL products to be purchased from every supplier while achieving the aspiration levels of multiple goals.	Environmental	Preemptive goal programming	Condition of returned products, variety of products from different suppliers, and quantity discounts offered by suppliers.
Kinoshita et al. [54, 55]	Minimizing the recycling cost and maximizing the recycling rate.	Environmental and economic	Goal programming	The ε constraint method and Goal Programming.
Igarashi et al. [56]	Designing a multi criteria disassembly system part selection and line balancing.	Environmental and economic	Integer Programming	Cost, recycling, and CO_2 savings.
Ghorabaee [57]	Allowing the DM to set the preferences to the MCDM algorithm.	Environmental and economic	Fuzzy liner physical programming	Implement the DM preferences in equations and apply it using MCDM algorithm.
Ilgin et al. [58]	Use of MCDM techniques in environmentally conscious manufacturing and product recovery.	Literature review	Literature review	Literature review.
Kalayci et al. [59]	Multi-objective fuzzy disassembly line balancing using a hybrid discrete artificial bee colony algorithm.	Environmental and economic	Hybrid discrete artificial bee colony algorithm	Lexicographic method, and fixed weighted method trying to optimize each conflicting concurrency.
Kalayci et al. [60]	Variable neighborhood search algorithm for disassembly lines.	Environmental and economic	Variable neighborhood search	Disassembly Line Balancing Problem and Sequence-Dependent Disassembly Line Balancing Problem.

Table 2.1. continued

Ondemir and Gupta [61]	Multi-criteria decision making model for advanced repair-to-order and disassembly-to-order system.	Environmental	Linear Physical Programming	Minimize the total cost and minimize the number of disposed items while reducing the uncertainty in products.
Ondemir et al. [62]	Optimal End-of-Life Management in Closed-Loop Supply Chains Using RFID and Sensors.	Environmental and economic	Mixed integer liner programming	RFID is considered as a support method to reduce the uncertainty in disassembly operations.
Vongbunyong et al. [63]	Basic behavior control of the vision-based cognitive robotic disassembly automation.	Economic	Cognitive robotics	Cognitive robot to perform the disassembly task.
Wang and Chan [64]	To demonstrate the applicability of AHP and propose methods for making an evaluation of remanufacturing alternatives.	Economic	Fuzzy hierarchical TOPSIS	Value (e.g. rare metal content, competition between imitated products, environmental impacts), cost involved, employee health and safety, and design difficulties.
ElSayed et al. [65]	To generalize the current models by accommodating an environment that is conducive to fuzzy problem solving.	Economic	Fuzzy linear physical programming and LPPW	Profit, monthly production level.
Joshi et al. [66]	To evaluate and select the best suppliers of multiple suppliers of the EOL products based on stated criteria for maximizing the profit, quality level, and material sales revenue and minimizing the disposal weight.	Economic and financial	Goal programming	The conditions of the EOL products, their collection costs and labor costs.

Table 2.1. continued

Ilgin and Gupta [67]	To present and discuss the development of research in Environmentally Conscious Manufacturing and Product Recovery (ECMPRO).	Literature review	Literature review	Literature review.
McGovern and Gupta [68]	To determine performance metrics for multiple objective end-of-life Disassembly Line Balancing Problem.	Economic	Preemptive goal programming	Early removal of hazardous parts, early removal of high- demand, and adjacent removal of parts with equivalent part removal directions.
William Ho [69]	To provide evidence that the integrated AHPs are better than the stand-alone AHP, and to aid the researchers and decision makers in applying the integrated AHPs effectively.	Literature review	Literature review	Literature review.
Rousis et al. [70]	To examine alternative scenarios/systems for WEEE management in Cyprus.	Environmental	MCDM: PROMETHEE	Performance and efficiency.
Valério et al. [71]	To present Multiple- criteria decision-making (MCDM) modeling for the selection of equipment suppliers in an automotive plant.	Financial and economic	Analytic Hierarchy Process	Cost, lead time, maintenance easiness, expected index of rejected products, yield and contamination.

Table 2.1 Review of related literature.

2.2 Design of Experiments (DOE) Literature Survey

Experimental design was first introduced in 1920s by R. A. Fischer, who developed the basic principles of factorial design and the associated data analysis known as ANOVA during research in improving the yield of agricultural crops [72]. Design of Experiment (DOE) has gained a wide interest 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 [73].

Aksoy and Gupta [74] have presented an efficient algorithm to determine the near-optimal buffer allocation of a given number of buffer slots in a remanufacturing cell with finite buffers and unreliable servers. The authors considered a manufacturing cell that consisted of three main modules: the disassembly and testing module for returned products, the disposition module for non-reusable returns, and the remanufacturing module. They have proposed a buffer allocation algorithm that distributed a given number of available buffer slots among the various stations to optimize the performance of the cell.

In the area of manufacturing, Kondapalli et al. [73] reviewed the literature on DOE techniques that have been employed for different welding processes. The paper also focused on the application of Taguchi method on fusion arc welding processes namely, gas tungsten arc welding and plasma arc welding.

For most disassembly systems, there are two crucial issues: one is the disassembly sequencing where the optimal or near optimal disassembly sequence determination is involved, and the other is the disassembly to order (DTO) where the number of end-of-life products to process is determined to fulfill the demand for specified numbers of components and materials.

For a good combination of those two issues, Ilgin and Tasoglu [75] have proposed a simulation optimization approach based on genetic algorithm (GA) for the simultaneous determination of disassembly sequence and disassembly-to-order decisions. The authors illustrated their proposed approach through a numerical example. Their study employed Taguchi's L9 orthogonal array experimental design to obtain the best values of GA parameters. This orthogonal array was designed with four factors with three levels.

Chang [76] presented a method that combined 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. The paper aimed at minimizing the cost of the filter, the filter losses, and the total harmonic distortion of currents and voltages at each bus simultaneously.

Mehmet Ali Ilgin at el. [77] studied the use of embedding sensors during their end-of-life (EOL) processing. They carried out separate design of experiments based on orthogonal arrays for conventional products (CPs) and sensor embedded products (SEPs). Detailed discrete event simulation models of both cases were developed by taking into consideration the precedence relationships among the components together with the routing of many different types of appliances through the line of disassembly. The study showed that the sensor embedded products SEPs not only decreased various costs and also increased revenue and profit.

Lazic and Mastorakis [78] studied the problem of testing black boxes and considered the combination of input parameters that affect an output parameter through an Orthogonal Array Testing Strategy (OATS). The authors analyzed software-system test requirements and corresponding models. Their study also presented a brief overview of the response surface methods (RSM) for computer experiments in the literature.

Finally, Moghaddam and Kolahan [79] proposed an approach that is based on Taguchi design matrix for the face milling process.

2.3 Task Allocation Literature Survey

Disassembly line balancing (DLB) is known as arranging of a group of tasks to an ordered sequence of stations for the purpose of optimizing performance. DLB problems optimize the disassembly line while meeting the demand for the parts retrieved from the returned products [80]. Several steps of recovery and remanufacturing are included in the disassembly process.

Metea et al. [81] provided a mathematical model for solving DLB problems through resource constraints. The authors aimed at minimizing the number of resources and workstations under determined cycle-times. The solution was obtained through GAMS-CPLEX.

Bentaha et al. [82] proposed a disassembly line balancing and sequencing problem for EOL products with hazardous parts. The authors aimed at maximizing the profit of the production line with uncertain task times. The tasks were arranged in a sequence of workstations while concurrently satisfying precedence and cycle time constraints. In order to cope with uncertainties, the authors developed an exact solution method based on integer programming and Monte Carlo sampling.

The disassembly line balancing problem has a profound effect since it is considered as one of the most efficient ways to achieve disassembly of large or largely produced products. In this regard, Güngör et al. [80] presented a heuristic to show the combination of several important factors in disassembly into the process of solving of a DLB problem.

The disassembly process includes a group of tasks that must be completed within a given time. Due to defects however, one or more tasks cannot be performed and leading to complications on the disassembly. To address these issues, Gungor and Gupta [83] discussed the disassembly line balancing problem in the presence of task failures (DLBP-F), and proposed an approach aiming at minimizing the impact of the defective part task assignments to workstations.

Torres et al [84] published a study for nondestructive automatic disassembly of personal computers where they considered a disassembly cell. The authors employed a computer vision system to recognize and localize the product and its components. An additional disassembly system responsible for generating the disassembly sequence and the planning of the disassembly movements was also proposed. The two systems cooperated with each other to provide semi-automatic disassembly operations.

Gutjahr and Nemhauser [85] first described a solution to the assembly line balancing problem with an algorithm that minimized the delays at each workstation based on the shortest route in a finite directed network. The proposed heuristic accounted for precedence relationships.

Erel and Gokcen [86] developed a modified version of an existing line-balancing problem algorithm. The proposed model was capable of considering any constraint that can be expressed as a function of task assignments.

McMullen and Tarasewich [87] used ant colony optimization techniques to solve the assembly line balancing problem with parallel workstations, stochastic task durations, and mixed models. Their methodology addressed several assembly line balancing problems.

The disassembly line balancing problem searches a sequence that targets the feasibility and minimization of number of workstations with reduced idle times. With this motivation, McGovern and Gupta [18] presented a genetic algorithm to obtain optimal or near-optimal solutions for disassembly line balancing problems.

Duta et al. [88] designed and balanced a disassembly line based on the equal piles approach to avoid uncertainties during the disassembly process. In addition, Duta and Filip [89] studied the line structure and proposed an algorithm that aimed at finding the best disassembly sequence. The authors concluded that their proposed algorithm provided a solution that improved the line balance.

ElSayed et al. [90] presented a genetic algorithm model to find the optimal disassembly sequence of a given product. The model provided reliable and quick input to the disassembly scheduling environments. The authors concluded that the multi-objective algorithm was practical and easy to use accounting for precedence relationships and additional constraints.

Gagnon and Morgan [91] conducted a review of the documented decisions and issues that explained the complications in disassembly line balancing problems.

Avikal et al. [92] proposed an efficient, near optimal, and a multi-criteria decision making technique based heuristics for assigning the disassembly tasks to the workstations. The PROMETHEE method was used to set the priorities of the assigned tasks. The authors concluded that the proposed technique helped in achieving substantial improvements in the performance compared with other heuristics.

2.4 Serverless Architecture Literature Survey

Serverless computing has recently gained considerable interest due to its powerful services, simple programming and deployment models, and efficient cost management. In spite of this trend in its adoption, the serverless architecture is still in its infancy and therefore the related work on using this architecture is scarce.

Serverless computing is preferred by highly scalable, event-driven applications since it deals with allocating resources as events arrive which can reduce the cost of pre-allocated or dedicated hardware.

McGrath and Brenner [93] have presented a novel serverless computing platform implemented in .NET and deployed in Microsoft Azure. The platform utilized windows containers as function execution environments. The authors also proposed metrics to evaluate the execution performance of serverless platforms and conduct tests on a prototype as well as AWS Lambda, Azure Functions, Google Cloud Functions, and IBM’s deployment of Apache Open Whisk. Their findings indicate that the prototype achieved greater throughput compared to other platforms in some aspects.

Conventionally, client-server-based video streaming systems are the most common video streaming systems. Ho and Lee [94] studied the problem of data recognition when growing a serverless video streaming system. The authors presented a new data reorganization algorithm that allowed a controllable tradeoff between data reorganization overhead and streaming load balance.

Lee and Leung [95] investigated a radically serverless architecture that relied on the client machines for distributed data storage and delivery. In this work, the authors developed fault-tolerance algorithms to maintain the stream delivery even if some clients failed.

Bolosky et al. [96] considered architecture for a serverless distributed file system that did not assume mutual trust among the client computers. The authors measured and analyzed a large set of client machines in a commercial environment to assess the feasibility of deploying this system on an existing desktop infrastructure.

Bolosky et al. [97] calculated results on disk usage, content, and file activity and also factored into their results machine uptimes, lifetimes, and loads. They concluded that the measured desktop infrastructure would possibly support their proposed system, providing availability on the order of one unfilled file request per user per thousand days.

Hendrickson et al. [98] proposed a new, open-source platform for building next-generation web services and applications in the burgeoning model of serverless computation called OpenLambda. The authors discussed the main aspects of serverless computation regarding the design and the implementation of such systems.

Bila et al. [99] presented their serverless architecture for securing Linux containers which provide continuous scanning for containers. The authors explored the design of an automated threat mitigation architecture based on OpenWhisk and Kubernetes.

Baldini et al. [100] investigated serverless functions and identified three competing constraints: functions should be considered as black boxes; function composition should obey a substitution principle with respect to synchronous invocation; and invocations should not be double-billed. They introduced the serverless trilemma, which captured the inherent tension between economics, performance, and synchronous composition.

Finally, Adya et al. [96] described serverless distributed file system and improved the performance through storing fewer copies of a file.

CHAPTER 3: RESEARCH PLAN

This section presents the model environment including the hardware setting and the developed algorithms to obtain the near-optimal/optimal disassembly sequence for a given EOL product. Figure 3.1 presents the proposed sensory system that includes an end-of-life personal computer, the robotic manipulator and the digital camera in addition to the captured image prior to disassembly.

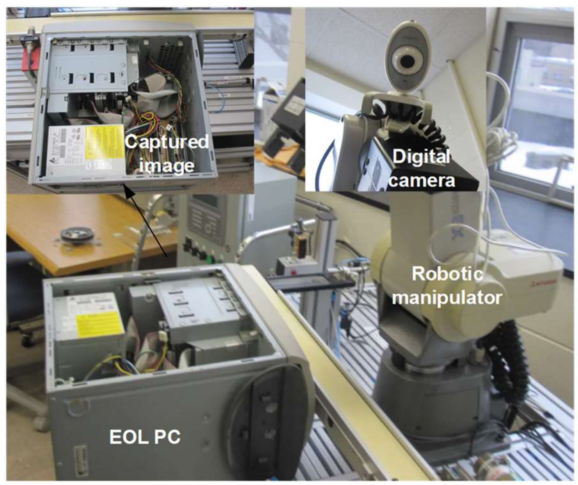


Figure 3.1 Sensory system: EOL PC, robotic manipulator, digital camera and the captured image prior to disassembly

Screenshot of the robot arm and the processor prior to the disassembly operation is presented in Figure 3.2.

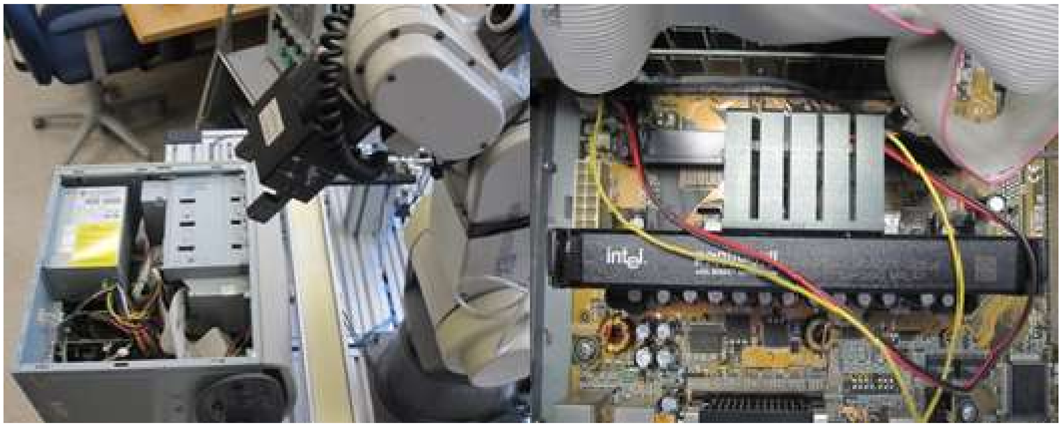


Figure 3.2 Screenshot of the robot arm and the processor prior to disassembly.

3.1 Disassembly Sequence Problem Mathematical Foundation

The MCDM algorithms use an evaluation function that would calculate the total time for disassembly operations. Equation (3.1) represents the formulation of the fitness function which comprises of three segments.

$$F_{sol} = \sum_{\forall i \in j} tt_{ij} + \sum_{\forall i \in j} mt_{ij} + \sum_{\forall i \in j} dt_{ij}. \quad (3.1)$$

The first part of equation (3.1) utilizes the distance formula to calculate the travel time of the robot arm between two components. This can be represented by equation (3.2).

$$tt_{ij} = \frac{\sqrt{\left(X_{k(j-1)} - X_{ij}\right)^2 + \left(Y_{k(j-1)} - Y_{ij}\right)^2 + \left(Z_{k(j-1)} - Z_{ij}\right)^2}}{sf}. \quad (3.2)$$

In this equation, X and Y represent respectively the width (x-axis), the height (y-axis) while Z denotes the depth (z-axis). The index i represents component i being disassembled in the jth sequence while index denotes the previously disassembled component in the (j-1)th sequence. The travel time between components i and k are calculated by dividing this distance by the average speed of the robot arm (sf). The robot speed is given as 7 centimeters per second (sf = 7 cm/sec), the average speed of the Mitsubishi Industrial Micro-Robot System Model RV-M1.

The second part of the fitness function introduces the time penalty for changing the disassembly method if present (Eq. 3.3). Here, the fitness function is penalized by 1 second for each method change, if any.

$$mt_{ij} = \begin{cases} 0, if\ no\ method\ change\ required \\ 1, if\ method\ change\ is\ required \end{cases}. \tag{3.3}$$

The third part represents the actual disassembly time required for component i in seconds. This can be expressed with equations below.

$$dt_{ij} = \{dt_{1j}, dt_{2j}, \dots, dt_{nj}\}, \qquad for\ i = 1, \dots, n. \tag{3.4}$$

In this work, the following values are used for the ten components (n = 10) in the EOL product.

$$dt_{ij} = \{2, 3, 3, 2, 3, 4, 2, 1, 3, 2\},\ for\ i = 1, \dots, 10. \tag{3.5}$$

In equation (3.1), the index boundaries are, $1 \leq i \leq n$, $1 \leq j \leq l$, and $0 \leq k \leq n$, where, n is the number of items in the EOL product structure and l is the length of the sequence generated by the simulated annealing algorithm in each run. Furthermore, the lower and upper boundaries for the sequence, $1 \leq l \leq n$, are naturally introduced into the fitness function, similar to the constraint, $i \neq k$. Zero values for both k and (j-1) indicate the initial position of the robot arm prior to disassembly when there is no sequence generated.

The algorithm is structured so that if two of the same-layer components are both made out the same material and are assigned destructive disassembly, a "pair" disassembly operation is conducted. In this case, these two components are disassembled concurrently with only a single penalty for the disassembly time. Please see [26] for the detailed description of the pairing logic.

CHAPTER 4: IMPLEMENTATION AND RESULTS

4.1 Implementation of the Tabu Search and Genetic Algorithm Models

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 (Figure 3.2) 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 subjected 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 non-destructive (ND) to destructive (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.

Figure 4.1 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 obtain the subsequent best solution. In the case where the current solution is not considered as a feasible one, the same iteration will be executed until a feasible (good) solution is obtained. These solutions will make sure that the algorithm will avoid trapping into 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 4.1 and the pseudo code for the overall search is given in Table 4.2.

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

Table 4.1.Tabu search algorithm.

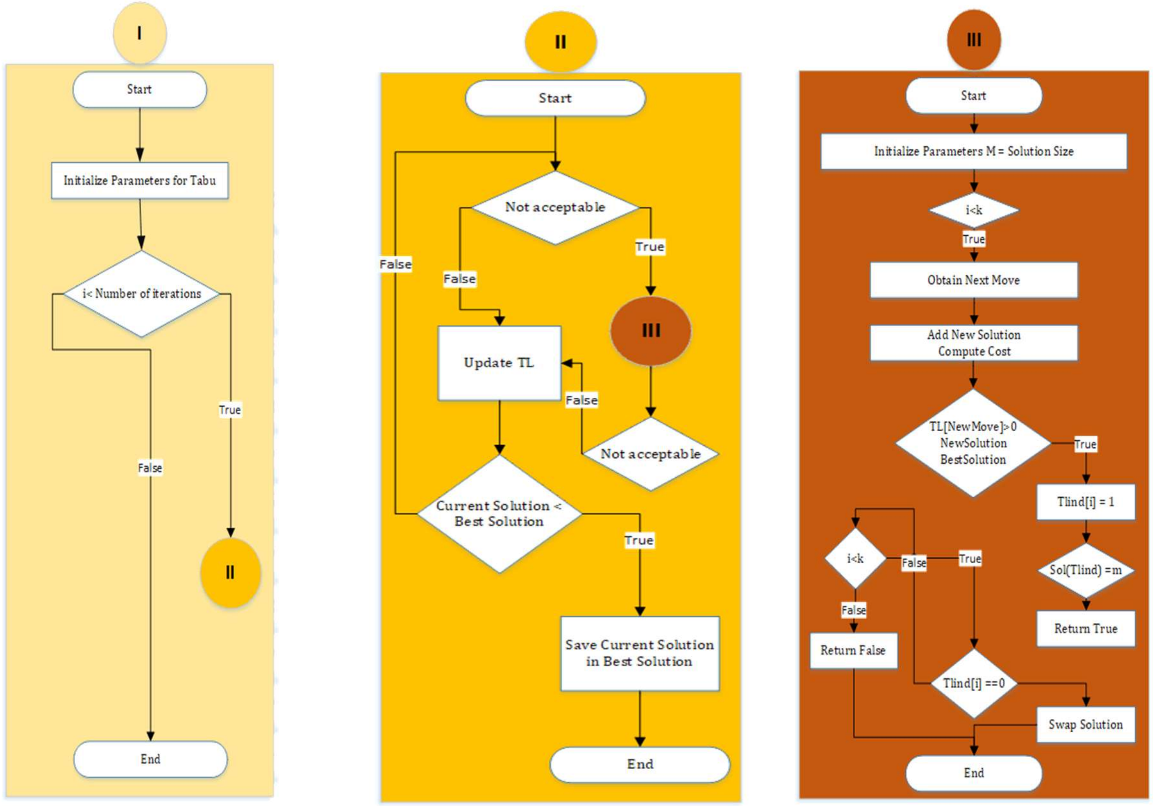


Figure 4.1.Tabu search algorithm flowchart.

```
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
```

Table 4.2 Pseudocode for Tabu search.

After initializing the algorithm parameters, the ComputeCost function will be executed to calculate the fitness for the first and initial solution, and then RunTS will iterate to find the optimal or near optimal solution. In the case where the next best feasible solution 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 4.2 demonstrates the Genetic Algorithm flowchart. GA parameters such as population, generation size and the number of iterations is initialized. This represent the call of GA functions such as Crossover, Permutation, and Chromosome.

Figure 4.3 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.

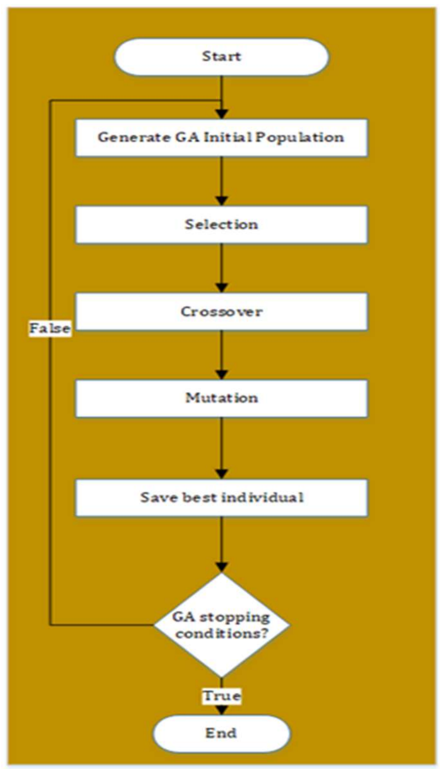


Figure 4.2 Genetic algorithm flowchart.

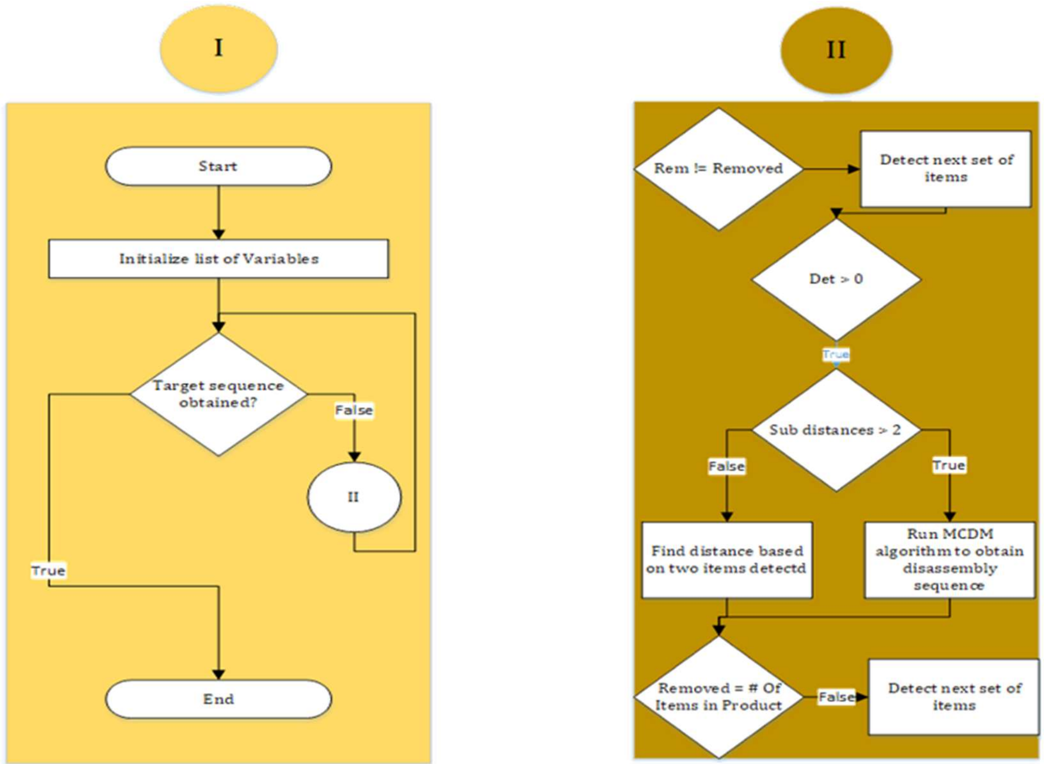


Figure 4.3 Genetic and Tabu search algorithms.

4.2 Implementation of the Tabu Search and Genetic Algorithm Models

This section presents a numerical example to demonstrate the functionally of the proposed methodology. Table 4.3 lists the components of the end-of-life product, material content of each product along with the required disassembly technique to recycle, reuse, store or properly dispose of these components. D denotes destructive disassembly whereas ND indicates that non-destructive disassembly method must be used.

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

Table 4.3 End-of-life product components, material content and required disassembly techniques.

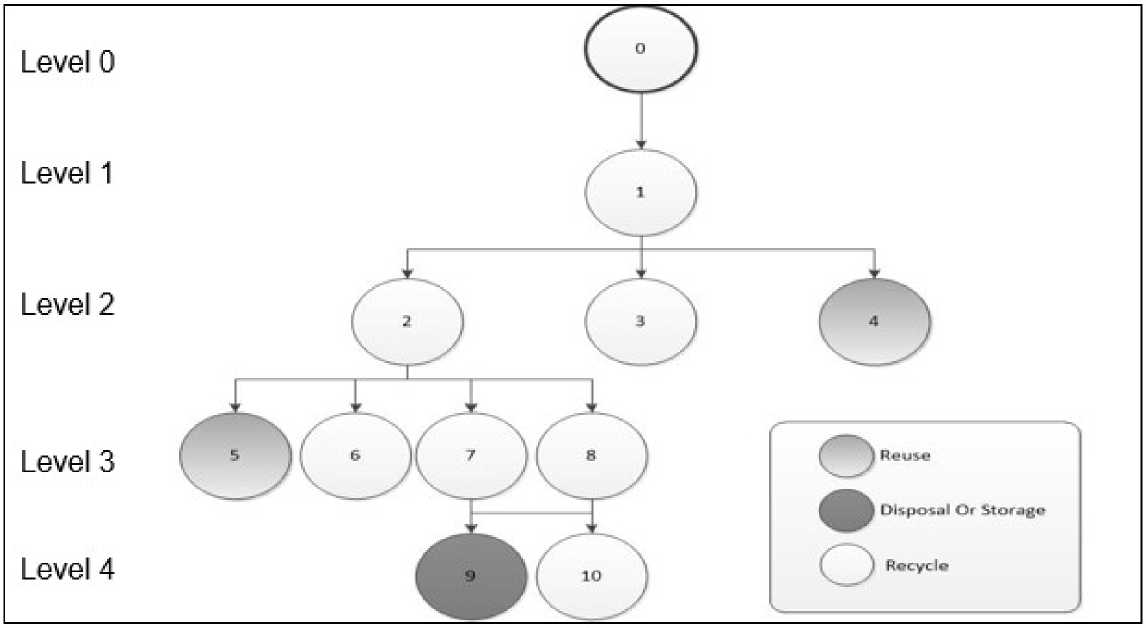


Figure 4.4 Bill-of-materials (BOM) for the EOL product.

The Tabu Search algorithm is applied to the numerical example provided in Table 4.3; for the product provided in Figure 4.4, 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 [22]. 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 4.4. 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).

	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

Table 4.4 Summary Statistics for Tabu search (TS) and Genetic algorithm (GA) run times in milliseconds.

Figure 4.5 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 Figure 4.6. The histograms indicate that Tabu Search (s2= 4133.405867) runs are more spread compared to Genetic Algorithm (s2 = 1267.214) runs. Figure 4.7 represent a side-by-side comparison for the same histogram represent in Figure 4.6, its clear from 4.7 when the data was presented on side by side that the mean for GA is higher than the mean of Tabu Search. Figure 4.7 explains that the run time required by GA is more than Tabu Search.

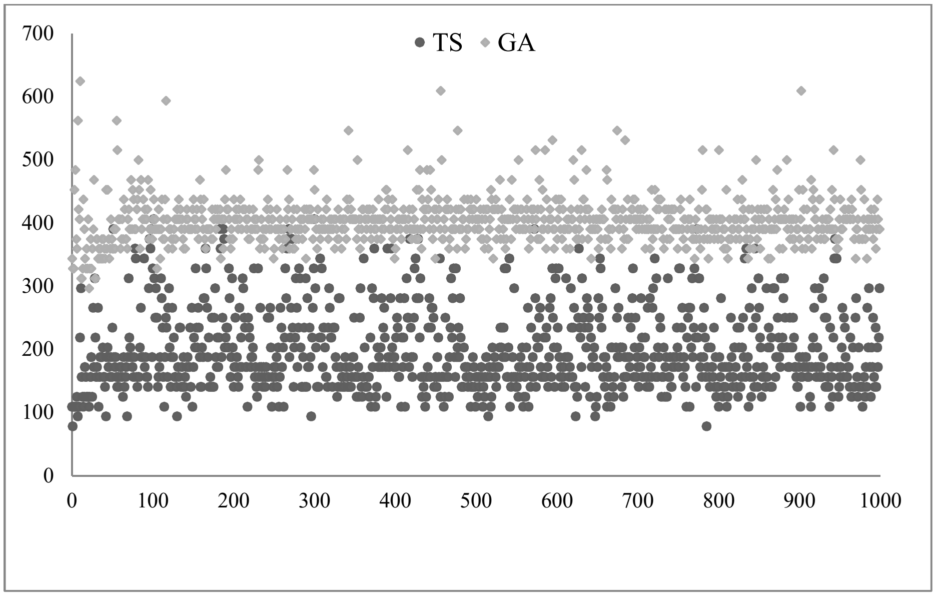


Figure 4.5 Scatter plots of Tabu search (TS) and Genetic algorithm (GA) run times in milliseconds.

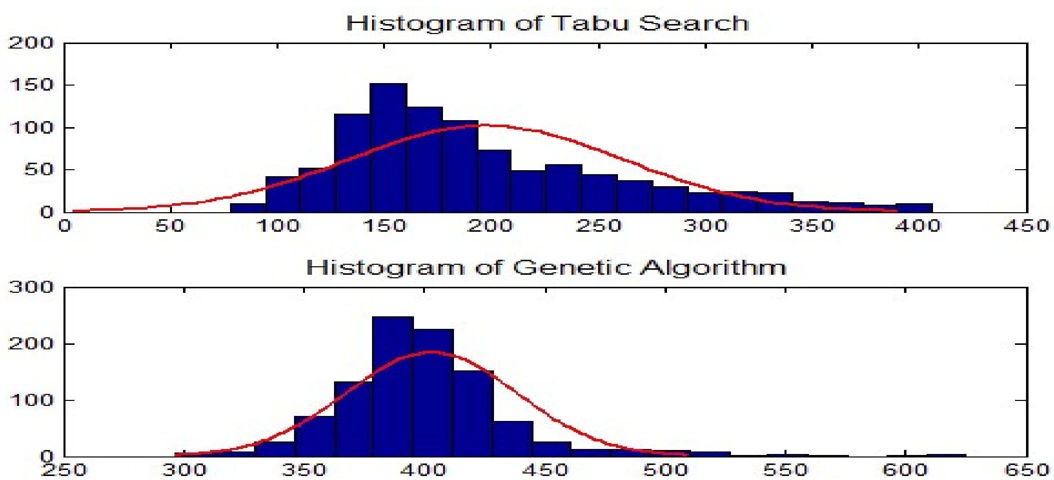


Figure 4.6 Histograms of Tabu search (TS) and Genetic algorithm (GA) run times in milliseconds.

Further distribution testing in the Arena simulation software indicated that both data sets were 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 data set 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 4.5). 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 4.6).

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 4.7, indicating that the variations between the data sets are significantly different.

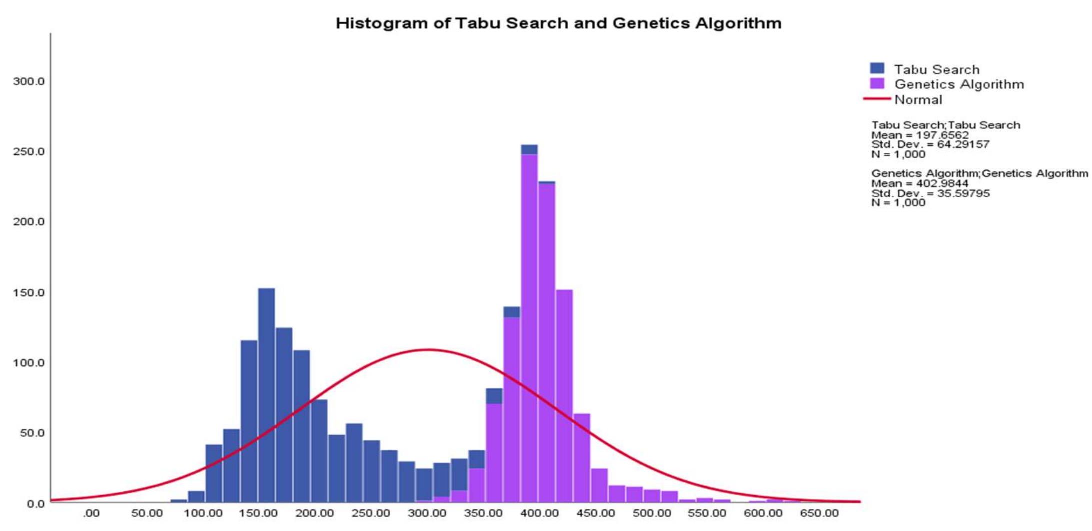


Figure 4.7 Histograms of Tabu search and Genetics algorithm.

	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 4.5 Kolmogorov-Smirnov and Shapiro-Wilk tests of normality.

	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 4.6 F-Test two-sample for variances 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				

Table 4.7 ANOVA: Single factor results.

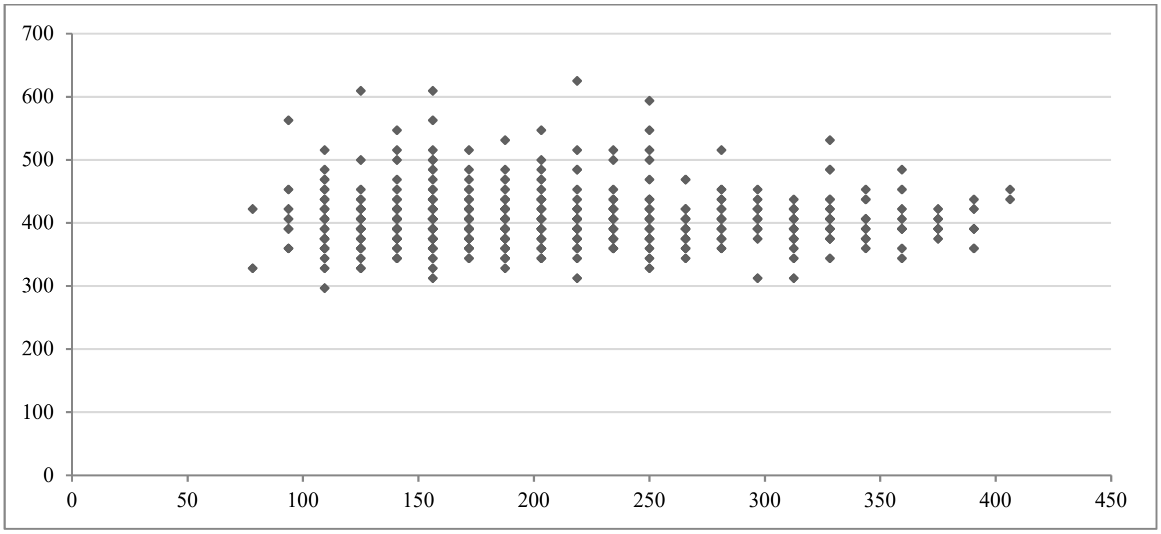


Figure 4.8 Scatter plot for Tabu search (TS) versus Genetic algorithm (GA) runs in milliseconds.

A scatter plot for Tabu Search (TS) versus Genetic Algorithm (GA) runs is plotted to illustrate the relationship between the two data sets (Figure 4.8).

In order to prove the samples are independent of each other, Pearson's 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 addition, we can say that 0.0121% (0.0112) of the variation in GA run times is explained by TS run times.

4.3 Implementation of the Analytical Hierarchy Process (AHP) and Simulated Annealing (SA) Models

In order to incorporate DMs' preferences into the process, AHP, a method capable of incorporating tangible and intangible factors into the model environment [1], has been utilized. AHP facilitates interaction with the model environment allowing DMs to assess and evaluate their decisions based on necessity [34]. AHP is a stepwise process using a numerical scale ranging from 1 to 9 to represent the DMs' preference for each activity [3]. Table 4.8 represents the original intensity of importance used in this study.

The levels of the AHP algorithm, namely, goal, criteria, and alternatives are provided in Figure 4.8. Here, the goal is to obtain the preference vector to decide on the appropriate EOL recovery option, viz., recycle, proper disposal, reuse, and storage. In the second level, each criterion is listed to include environmental, economic and social considerations imposed by the decision maker(s). A list of all alternatives is provided in the third level of the hierarchy.

Intensity of importance	Definition	Description
1	Equal importance	Two activities contribute equally to the objective
3	Weak importance of one over another	Experience and judgement slightly favor one activity over anothe
5	Essential or strong importance	Experience and judgment strongly favor one activity over another
7	Demonstrated importance	An activity is strongly favored and its dominance is demonstrated in practice
9	Absolute importance	The evidence favoring one activity over another is of the highest possible order of affirmation
2 ,4 ,6 ,8	Intermediate values between the two adjacent judgments	When compromise is needed
Reciprocals of above nonzero	If activity *i* has one of the above nonzero numbers assigned to it when compared with activity *j*, then *j* has the reciprocal value when compared with *i*.	

Table 4.8 Verbal and numerical scale representation for AHP [32].

The EOL product recovery system is initiated by the input provided by the decision maker. Based on the Bill-of-Materials of the end-of-life product, the decision maker sets the preference levels using Table 4.9.

Table 4.9 represents the preferences for each component. In the EOL section of the component is the preferences to select between Environmental Economic, and Social. The second set of preferences to choose between Reuse, Recycling, Proper Disposal, and Storage.

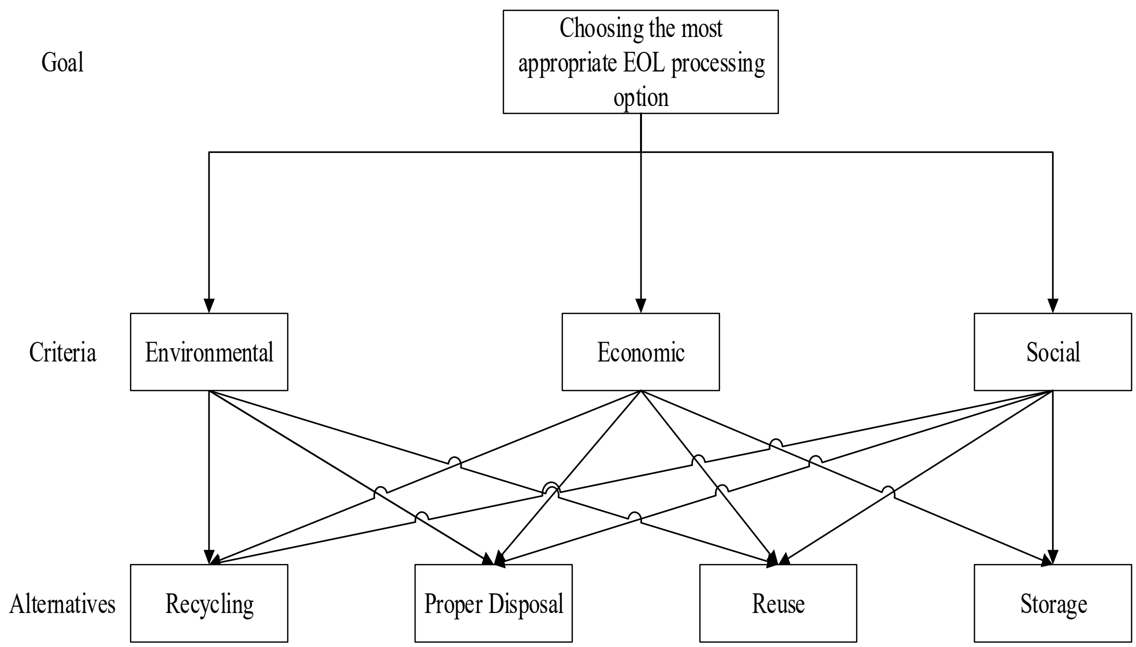


Figure 4.9 AHP priority vector generation process (Adopted from [3]).

The preference vector is then generated based on these preference assignments. The flexibility ratio is a percentage to be set by the DM and represents the stringency of the DM input. For instance, if the flexibility ratio is set as five percent, this will imply that if the difference between the destructive and non-destructive method is less than or equal to this value, the simulation will select the appropriate disassembly method to reduce the overall time. Otherwise, the disassembly method will be selected based on the DMs' preference.

The weight vector is then calculated to represent the probability of each criterion using equation (4.1).

$$PM_{pq} = \begin{pmatrix} 1 & \cdots & w_r \\ \vdots & \ddots & \vdots \\ 1/w_0 & \cdots & 1 \end{pmatrix} = \begin{cases} 1, \forall\, p,q \in 1,2,\ldots .r \; where\; p = q \\ w_{pq}, \forall\, p,q \in 1,2,\ldots .r \; where\; p < q \\ 1/w_{qp}, \forall\, p,q \in 1,2,\ldots .r \; where\; p > q \end{cases}. \tag{4.1}$$

Once the pairwise comparison matrix is built, the next step is to normalize this matrix by summing up each column and dividing the column cells by the summation:

$$NPM_{pq} = \frac{w_{pq}}{\sum_{p=1}^{r} w_{pq}} \forall\, p,q \,\epsilon\, 1,\ldots,r. \tag{4.2}$$

This process will generate a priority vector that will be used to decide on the disassembly method and the EOL option for each component:

$$PPM_q = \frac{\sum_{q=1}^{r} NV_{pq}}{r} * 100 \;\forall\, p,q \,\epsilon\, 1,\ldots,r. \tag{4.3}$$

The variable *r* is bounded by 1 and the number of criteria and the number of EOL processing options based on the AHP problem structure. In this study, *r* = 1,….3, in the first step and *r* = 1,…, 4, in the second.

At every step, the consistency ratio (*CR*) is calculated to ensure the consistency of corresponding pairwise comparison matrix, i.e., judgment matrix. If the resulting *CR* value is higher than 10%, the decision maker is asked to revise the corresponding set of parameters. This is continued until the consistency ratio is less than or equal to 10%.

Following the preference vector calculations, the normalized decision vector values are fed into the Simulated Annealing (SA) search algorithm. The algorithm is then utilized to generate the optimum or near-optimum solution, based on the DM preferences. Schematic representation of the decision maker-centered EOL product disassembly sequencing infrastructure is provided in Figure 4.9.

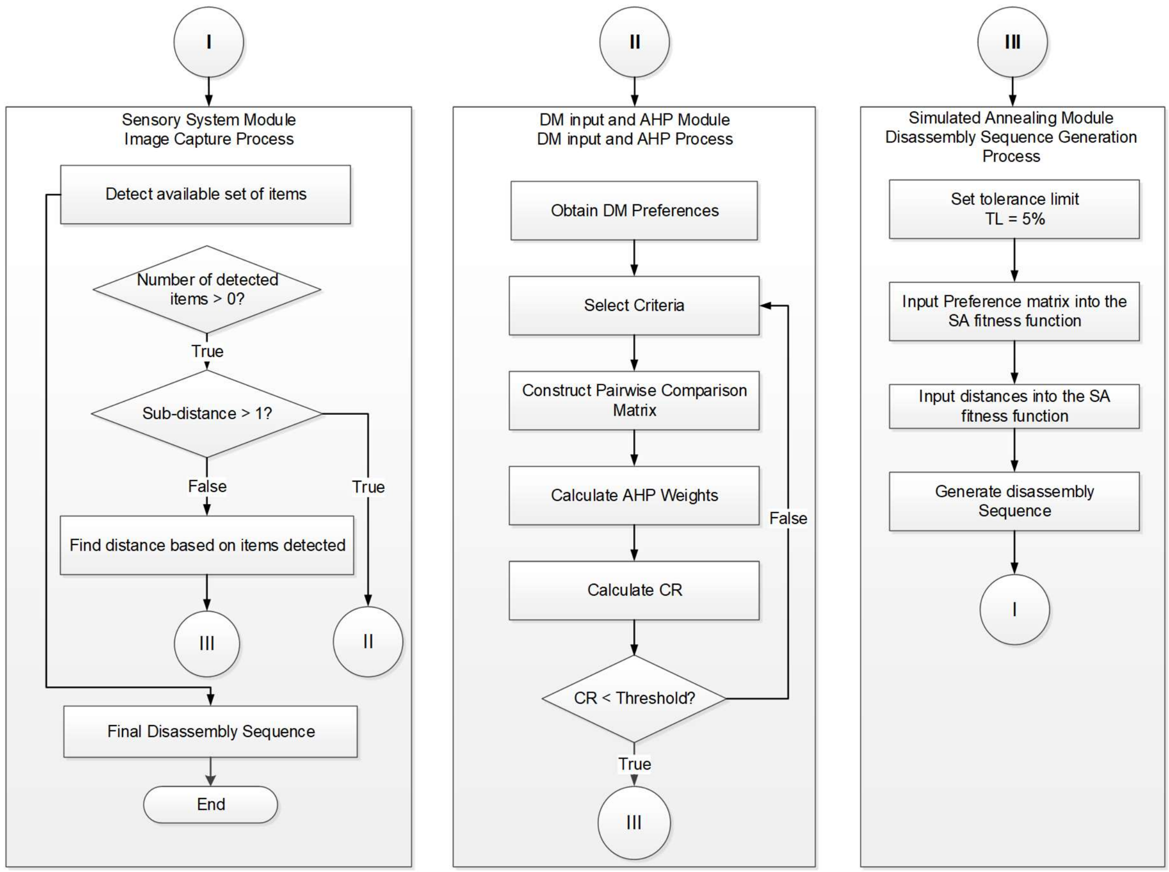


Figure 4.10 Decision maker-centered EOL product disassembly sequencing infrastructure: sensory system, AHP and SA modules.

Component 1	**EOL**	Environmental	Economic	5/1	Environmental	Social	3/1
		Economic	Social	7/1			
	Environmental	Reuse	Recycling	1/5	Recycling	Storage	5/1
		Reuse	Storage	1/1	Recycling	Proper Disposal	7/1
		Reuse	Proper Disposal	1/3	Storage	Proper Disposal	1/3
	Economic	Reuse	Recycling	1/7	Recycling	Storage	7/1
		Reuse	Storage	1/1	Recycling	Proper Disposal	9/1
		Reuse	Proper Disposal	1/5	Storage	Proper Disposal	1/5
	Social	Reuse	Recycling	1/9	Recycling	Storage	9/1
		Reuse	Storage	1/1	Recycling	Proper Disposal	7/1
		Reuse	Proper Disposal	1/7	Storage	Proper Disposal	1/7
Component 2	**EOL**	Environmental	Economic	1/7	Environmental	Social	3/1
		Economic	Social	1/5			
	Environmental	Reuse	Recycling	1/5	Recycling	Storage	5/1
		Reuse	Storage	1/1	Recycling	Proper Disposal	7/1
		Reuse	Proper Disposal	1/3	Storage	Proper Disposal	1/3
	Economic	Reuse	Recycling	1/5	Recycling	Storage	5/1
		Reuse	Storage	1/1	Recycling	Proper Disposal	7/1
		Reuse	Proper Disposal	1/3	Storage	Proper Disposal	1/3
	Social	Reuse	Recycling	1/5	Recycling	Storage	5/1
		Reuse	Storage	1/1	Recycling	Proper Disposal	7/1
		Reuse	Proper Disposal	1/3	Storage	Proper Disposal	1/3

Component 6	**EOL**	Criteria A	Criteria B	Preference	Criteria A	Criteria B	Preference
		Environmental	Economic	1/3	Environmental	Social	1/5
		Economic	Social	1.00			
	Environmental	Reuse	Recycling	3/1	Recycling	Storage	1/5
		Reuse	Storage	0.20	Recycling	Proper Disposal	1.00
		Reuse	Proper Disposal	3.00	Storage	Proper Disposal	5.00
	Economic	Criteria A	Criteria B	Preference	Criteria A	Criteria B	Preference
		Reuse	Recycling	3/1	Recycling	Storage	1/5
		Reuse	Storage	1/5	Recycling	Proper Disposal	1/1
		Reuse	Proper Disposal	3/1	Storage	Proper Disposal	5/1
	Social	Criteria A	Criteria B	Preference	Criteria A	Criteria B	Preference
		Reuse	Recycling	3.00	Recycling	Storage	1/5
		Reuse	Storage	0.20	Recycling	Proper Disposal	1/1
		Reuse	Proper Disposal	3.00	Storage	Proper Disposal	5/1
Component 7	**EOL**	Environmental	Economic	3.00	Environmental	Social	1.00
		Economic	Social	5.00			
	Environmental	Reuse	Recycling	5/1	Recycling	Storage	1/5
		Reuse	Storage	3/1	Recycling	Proper Disposal	1/1
		Reuse	Proper Disposal	7/1	Storage	Proper Disposal	3/1
	Economic	Reuse	Recycling	1/3	Recycling	Storage	5/1
		Reuse	Storage	1/1	Recycling	Proper Disposal	5/1
		Reuse	Proper Disposal	1/3	Storage	Proper Disposal	1/3
	Social	Reuse	Recycling	1/3	Recycling	Storage	5/1
		Reuse	Storage	1/1	Recycling	Proper Disposal	5/1
		Reuse	Proper Disposal	1/3	Storage	Proper Disposal	1/3

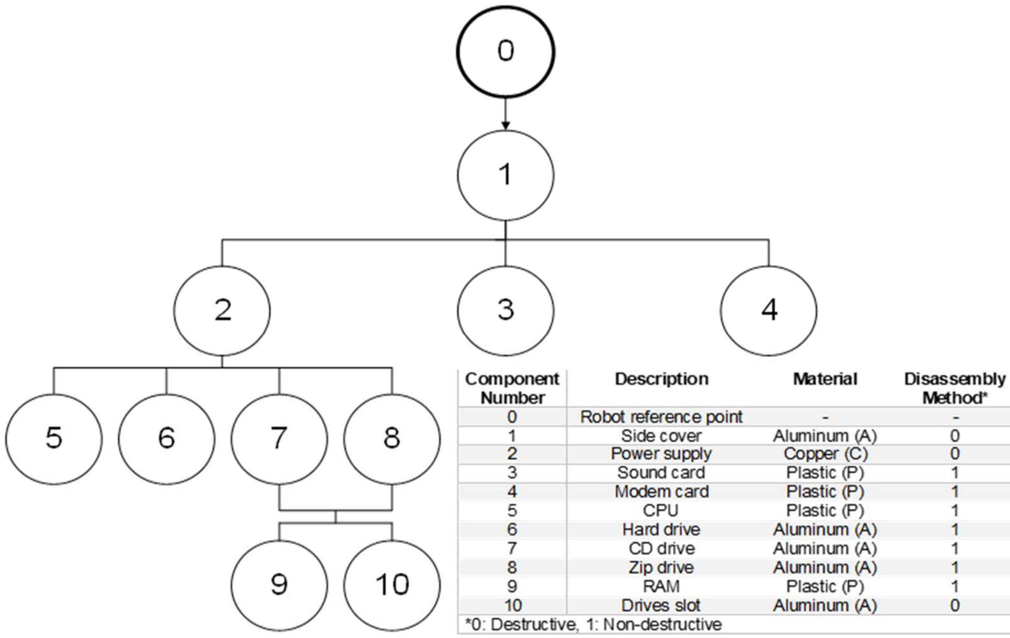
Component 3	EOL	Environmental	Economic	0.33	Environmental	Social	5.00
		Economic	Social	3.00			
	Environmental	Reuse	Recycling	7.00	Recycling	Storage	1/3
		Reuse	Storage	5.00	Recycling	Proper Disposal	1.00
		Reuse	Proper Disposal	3.00	Storage	Proper Disposal	3.00
	Economic	Reuse	Recycling	7.00	Recycling	Storage	0.33
		Reuse	Storage	5.00	Recycling	Proper Disposal	1.00
		Reuse	Proper Disposal	3.00	Storage	Proper Disposal	3.00
	Social	Reuse	Recycling	7/1	Recycling	Storage	1/3
		Reuse	Storage	5/1	Recycling	Proper Disposal	1/1
		Reuse	Proper Disposal	3/1	Storage	Proper Disposal	3/1
Component 4	EOL	Environmental	Economic	1/3	Environmental	Social	5/1
		Economic	Social	3/1			
	Environmental	Reuse	Recycling	7/1	Recycling	Storage	1/3
		Reuse	Storage	5/1	Recycling	Proper Disposal	1/1
		Reuse	Proper Disposal	3/1	Storage	Proper Disposal	3/1
	Economic	Reuse	Recycling	7/1	Recycling	Storage	1/3
		Reuse	Storage	5/1	Recycling	Proper Disposal	1/1
		Reuse	Proper Disposal	3/1	Storage	Proper Disposal	3/1
	Social	Reuse	Recycling	7/1	Recycling	Storage	1/3
		Reuse	Storage	5/1	Recycling	Proper Disposal	1/1
		Reuse	Proper Disposal	3/1	Storage	Proper Disposal	3/1
Component 5	EOL	Environmental	Economic	1/1	Environmental	Social	1/5
		Economic	Social	1/3			
	Environmental	Reuse	Recycling	1/5	Recycling	Storage	1/5
		Reuse	Storage	1/1	Recycling	Proper Disposal	7/1
		Reuse	Proper Disposal	1/3	Storage	Proper Disposal	1/3
	Economic	Reuse	Recycling	7/1	Recycling	Storage	1/7
		Reuse	Storage	1/1	Recycling	Proper Disposal	9/1
		Reuse	Proper Disposal	1/5	Storage	Proper Disposal	5/1
	Social	Reuse	Recycling	1/5	Recycling	Storage	5/1
		Reuse	Storage	1/1	Recycling	Proper Disposal	3/1
		Reuse	Proper Disposal	1/3	Storage	Proper Disposal	1/3
Component 8	EOL	Environmental	Economic	1/1	Environmental	Social	1/5
		Economic	Social	1/3			
	Environmental	Reuse	Recycling	3/1	Recycling	Storage	1/3
		Reuse	Storage	5/1	Recycling	Proper Disposal	1/1
		Reuse	Proper Disposal	7/1	Storage	Proper Disposal	3/1
	Economic	Reuse	Recycling	3/1	Recycling	Storage	1/3
		Reuse	Storage	5/1	Recycling	Proper Disposal	1/1
		Reuse	Proper Disposal	7/1	Storage	Proper Disposal	3/1
	Social	Reuse	Recycling	3/1	Recycling	Storage	1/3
		Reuse	Storage	5/1	Recycling	Proper Disposal	1/1
		Reuse	Proper Disposal	7/1	Storage	Proper Disposal	3/1
Component 9	EOL	Environmental	Economic	5/1	Environmental	Social	1/3
		Economic	Social	3/1			
	Environmental	Reuse	Recycling	5/1	Recycling	Storage	1/5
		Reuse	Storage	3/1	Recycling	Proper Disposal	1/3
		Reuse	Proper Disposal	3/1	Storage	Proper Disposal	3/1
	Economic	Reuse	Recycling	5/1	Recycling	Storage	1/5
		Reuse	Storage	3/1	Recycling	Proper Disposal	1/3
		Reuse	Proper Disposal	3/1	Storage	Proper Disposal	3/1
	Social	Reuse	Recycling	5/1	Recycling	Storage	1/5
		Reuse	Storage	3/1	Recycling	Proper Disposal	1/3
		Reuse	Proper Disposal	3/1	Storage	Proper Disposal	3/1
Component 10	EOL	Environmental	Economic	3/1	Environmental	Social	1/5
		Economic	Social	1/3			
	Environmental	Reuse	Recycling	3/1	Recycling	Storage	1/5
		Reuse	Storage	1/5	Recycling	Proper Disposal	1/1
		Reuse	Proper Disposal	3/1	Storage	Proper Disposal	5/1
	Economic	Reuse	Recycling	3/1	Recycling	Storage	1/5
		Reuse	Storage	1/5	Recycling	Proper Disposal	1/1
		Reuse	Proper Disposal	3/1	Storage	Proper Disposal	5/1
	Social	Reuse	Recycling	3/1	Recycling	Storage	1/5
		Reuse	Storage	1/5	Recycling	Proper Disposal	1/1
		Reuse	Proper Disposal	3/1	Storage	Proper Disposal	5/1

Table 4.9 Decision maker parameters and sample data

The flowchart shown in Figure 4.10 describes the process of disassembly sequence generation using the two steps discussed previously. In the first step, AHP is employed to generate the DM preferences and to obtain to the weight matrices. In the second step, SA is utilized to obtain the optimum or near-optimum solution using the provided DM preferences. In order to take into account the uncertainty in the EOL product structure, a component discovery operation is conducted prior to each disassembly process.

This section details the integrated disassembly sequencing modules and demonstrates the functionality of the AHP and SA algorithms via a numerical example.

The hierarchy of the EOL product used in this study along with its component and material contents is provided in Figure 4.11. As illustrated in the figure, the EOL product is composed of ten components.



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

*0: Destructive, 1: Non-destructive

Figure 4.11 Schematic representation of the EOL product.

The steps of the EOL product disassembly sequencing system is provided below.

Step 1. Detect all available objects for disassembly using the camera and the sensory system (Figure 3.1 and Figure 3.2).

Step 2. Obtain decision maker input. From the DM preferences select the preferences related to the objects detected in Step 1. For instance, if the components 2, 3 and 4 have been detected in the first round, the corresponding weight vectors provided below are then calculated to represent the probability of each vector using equation (4.

Using the weight vectors, the pairwise matrix is then generated using equation (4.2). The pairwise matrix is calculated in two iterations. The first iteration is to decide between the criteria Environmental (en), Economic (ec), and Social (so).

$$PM_2(en, ec, so\,) = \begin{pmatrix} 1 & \frac{1}{7} & \frac{1}{5} \\ 7 & 1 & 3 \\ 5 & \frac{1}{3} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0.14 & 0.20 \\ 7 & 1 & 3 \\ 5 & 0.33 & 1 \end{pmatrix},$$

$$PM_3(en, ec, so\,) = \begin{pmatrix} 1 & \frac{1}{3} & 3 \\ 3 & 1 & 5 \\ \frac{1}{3} & \frac{1}{5} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0.33 & 3 \\ 3 & 1 & 5 \\ 0.33 & 0.20 & 1 \end{pmatrix},$$

$$PM_4(sn, ec, so\,) = \begin{pmatrix} 1 & \frac{1}{3} & 3 \\ 3 & 1 & 5 \\ \frac{1}{3} & \frac{1}{5} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0.33 & 3 \\ 3 & 1 & 5 \\ 0.33 & 0.20 & 1 \end{pmatrix}.$$

The pairwise matrices are then normalized using equation (4.3). The criterion which is selected is marked in bold.

$$PM_2(en, ec, so\,) = \begin{pmatrix} \frac{1}{13} & \frac{0.14}{1.47} & \frac{0.20}{4.20} \\ \frac{7}{13} & \frac{1}{1.47} & \frac{3}{4.20} \\ \frac{5}{13} & \frac{0.33}{1.47} & \frac{1}{4.20} \end{pmatrix} = \begin{pmatrix} 0.077 & 0.097 & 0.048 \\ 0.538 & 0.677 & 0.714 \\ 0.384 & 0.226 & 0.238 \end{pmatrix} =$$

$$\begin{pmatrix} 0.221 \\ 1.930 \\ 0.848 \end{pmatrix} = \begin{pmatrix} \frac{0.221}{3} \\ \frac{1.930}{3} \\ \frac{0.848}{3} \end{pmatrix} = \begin{pmatrix} 7.30 \\ \mathbf{64.3} \\ 28.2 \end{pmatrix},$$

$$PM_3(en, ec, so\,) = \begin{pmatrix} \frac{1}{4.33} & \frac{0.33}{1.53} & \frac{3}{9} \\ \frac{3}{4.33} & \frac{1}{1.53} & \frac{5}{9} \\ \frac{0.33}{4.33} & \frac{0.20}{1.53} & \frac{1}{9} \end{pmatrix} = \begin{pmatrix} 0.077 & 0.097 & 0.048 \\ 0.538 & 0.677 & 0.714 \\ 0.384 & 0.226 & 0.238 \end{pmatrix} =$$

$$\begin{pmatrix} 0.221 \\ 1.930 \\ 0.848 \end{pmatrix} = \begin{pmatrix} \frac{0.221}{3} \\ \frac{1.930}{3} \\ \frac{0.848}{3} \end{pmatrix} = \begin{pmatrix} 7.30 \\ \mathbf{64.3} \\ 28.2 \end{pmatrix},$$

$$PM_4(en, ec, so\,) = \begin{pmatrix} 1 & 0.33 & 3 \\ 3 & 1 & 5 \\ 0.33 & 0.20 & 1 \end{pmatrix} = \begin{pmatrix} 0.077 & 0.097 & 0.048 \\ 0.538 & 0.677 & 0.714 \\ 0.384 & 0.226 & 0.238 \end{pmatrix} =$$

$$\begin{pmatrix} 0.221 \\ 1.930 \\ 0.848 \end{pmatrix} = \begin{pmatrix} \frac{0.221}{3} \\ \frac{1.930}{3} \\ \frac{0.848}{3} \end{pmatrix} = \begin{pmatrix} 7.30 \\ \mathbf{64.3} \\ 28.2 \end{pmatrix}.$$

The second iteration uses the results obtained from the first iteration and selects the proper pairwise matrix to generate the EOL processing option for each component. As explained previously, the EOL processing options include reuse (ru), recycling (rc), storage (st), and proper disposal (pd).

$$PM_2(ru, rc, st, pd) = \begin{pmatrix} 1 & \frac{1}{5} & 1 & \frac{1}{3} \\ 5 & 1 & 5 & 7 \\ 1 & \frac{1}{5} & 1 & \frac{1}{3} \\ 3 & \frac{1}{7} & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0.20 & 1 & 0.33 \\ 5 & 1 & 5 & 7 \\ 1 & 0.20 & 1 & 0.33 \\ 3 & 0.14 & 3 & 1 \end{pmatrix},$$

$$PM_3(ru, rc, st, pd) = \begin{pmatrix} 1 & 7 & 5 & 3 \\ \frac{1}{7} & 1 & \frac{1}{3} & 1 \\ \frac{1}{5} & 3 & 1 & 3 \\ \frac{1}{3} & 1 & \frac{1}{3} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 7 & 5 & 3 \\ 0.14 & 1 & 0.33 & 1 \\ 0.20 & 3 & 1 & 3 \\ 0.33 & 1 & 0.33 & 1 \end{pmatrix},$$

$$PM_4(ru, rc, st, pd) = \begin{pmatrix} 1 & 7 & 5 & 3 \\ \frac{1}{7} & 1 & \frac{1}{3} & 1 \\ \frac{1}{5} & 3 & 1 & 3 \\ \frac{1}{3} & 1 & \frac{1}{3} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 7 & 5 & 3 \\ 0.14 & 1 & 0.33 & 1 \\ 0.20 & 3 & 1 & 3 \\ 0.33 & 1 & 0.33 & 1 \end{pmatrix}.$$

The pairwise matrices are then normalized using equation (4.3).

$$PM_2(ru,rc,st,pd) = \begin{pmatrix} 1 & 0.20 & 1 & 0.33 \\ 5 & 1 & 5 & 7 \\ 1 & 0.20 & 1 & 0.33 \\ 3 & 0.14 & 3 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{10} & \frac{0.20}{1.54} & \frac{1}{10} & \frac{0.33}{8.67} \\ \frac{5}{10} & \frac{1}{1.54} & \frac{5}{10} & \frac{7}{8.67} \\ \frac{1}{10} & \frac{0.2}{1.54} & \frac{1}{10} & \frac{0.33}{8.67} \\ \frac{3}{10} & \frac{0.14}{1.54} & \frac{3}{10} & \frac{1}{8.67} \end{pmatrix} =$$

$$\begin{pmatrix} 0.1 & 0.129 & 0.1 & 0.038 \\ 0.5 & 0.648 & 0.5 & 0.807 \\ 0.1 & 0.129 & 0.1 & 0.038 \\ 0.3 & 0.092 & 0.3 & 0.115 \end{pmatrix} = \begin{pmatrix} 0.092 \\ 0.613 \\ 0.092 \\ 0.201 \end{pmatrix} = \begin{pmatrix} 9.2 \\ \mathbf{61.3} \\ 9.2 \\ 20.1 \end{pmatrix},$$

$$PM_3(ru,rc,st,pd) = \begin{pmatrix} 1 & 7 & 5 & 3 \\ 0.14 & 1 & 0.33 & 1 \\ 0.20 & 3 & 1 & 3 \\ 0.33 & 1 & 0.33 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{1.68} & \frac{7}{12} & \frac{5}{6.67} & \frac{3}{8} \\ \frac{0.14}{1.68} & \frac{1}{12} & \frac{0.33}{6.67} & \frac{1}{8} \\ \frac{0.20}{1.68} & \frac{3}{12} & \frac{1}{6.67} & \frac{3}{8} \\ \frac{0.33}{1.68} & \frac{1}{12} & \frac{0.33}{6.67} & \frac{1}{8} \end{pmatrix} =$$

$$\begin{pmatrix} 0.596 & 0.583 & 0.75 & 0.375 \\ 0.085 & 0.083 & 0.05 & 0.125 \\ 0.119 & 0.25 & 0.15 & 0.375 \\ 0.198 & 0.083 & 0.05 & 0.125 \end{pmatrix} = \begin{pmatrix} 0.576 \\ 0.085 \\ 0.223 \\ 0.114 \end{pmatrix} = \begin{pmatrix} \mathbf{57.6} \\ 8.50 \\ 22.3 \\ 11.4 \end{pmatrix},$$

$$PM_4(ru,rc,st,pd) = \begin{pmatrix} 1 & 7 & 5 & 3 \\ 0.14 & 1 & 0.33 & 1 \\ 0.20 & 3 & 1 & 3 \\ 0.33 & 1 & 0.33 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{1.68} & \frac{7}{12} & \frac{5}{6.67} & \frac{3}{8} \\ \frac{0.14}{1.68} & \frac{1}{12} & \frac{0.33}{6.67} & \frac{1}{8} \\ \frac{0.20}{1.68} & \frac{3}{12} & \frac{1}{6.67} & \frac{3}{8} \\ \frac{0.33}{1.68} & \frac{1}{12} & \frac{0.33}{6.67} & \frac{1}{8} \end{pmatrix} =$$

$$\begin{pmatrix} 0.596 & 0.583 & 0.75 & 0.375 \\ 0.085 & 0.083 & 0.05 & 0.125 \\ 0.119 & 0.25 & 0.15 & 0.375 \\ 0.198 & 0.083 & 0.05 & 0.125 \end{pmatrix} = \begin{pmatrix} 0.576 \\ 0.085 \\ 0.223 \\ 0.114 \end{pmatrix} = \begin{pmatrix} \mathbf{57.6} \\ 8.50 \\ 22.3 \\ 11.4 \end{pmatrix}.$$

Table 4.12 shows the user preference input, AHP pairwise comparison matrices, priority vectors, and the consistency ratios for all components in the EOL product. As it can be observed from the table, the DM preference vector indicates the percentages of destructive and non-destructive disassembly operations along with the percentages of components to be recycled, reused, stored and disposed of for each component in the EOL product. These values are then used in the fitness function calculations.

After assigning the DM preferences, the Simulated Annealing (SA) search algorithm is applied to generate the optimum or near-optimum solution, based on the DM preferences. The SA algorithm is explained in detail in Step 3.

Step 3. Introduce the preference percentages into the disassembly matrix to calculate the disassembly sequence via the simulated annealing algorithm. Here, since there is only one item (component 1) detected in the initial step, the AHP algorithm is immediately executed without the simulated annealing search to generate the preference vector for the first item in the EOL product structure.

In the second iteration, three components are detected (components 2, 3 and 4). Since there is more than one component in this step, following the preference matrix calculations, the SA search algorithm is initiated to generate the optimum or near-optimum disassembly sequence. Table 4.10 represents the results of the sub-matrices used in this step.

Iteration	**Detected Component**	**X-axis**	**Y-axis**	**Z-axis**	**Purpose**	**Method**	**Material**	**Disassembly Time**
1	0	0	0	0	0	1	0	0
	2	42	80.5	110.0	0	1	1	3
2	3	127	89	210.0	1	0	2	3
	4	146.333	90.333	210.0	1	0	2	2

Table 4.10 Sample run results

Here, the disassembly method is categorized as destructive and non-destructive and are represented by values 0 and 1, respectively. Similarly, 0, 1, 2, and 3 indicate the purpose of the disassembly, viz.; reuse, recycle, storage, and proper disposal, respectively.

In the first simulated annealing solution provided in Table 4.10, the order of the components (0, 2, 3, 4) represents the disassembly sequence where "0" is the robot arm reference point followed by the component indices. Using the SA fitness function given in equation 5, the fitness value is then calculated as 49.1646115780334. Table 4.11 shows the disassembly sequence generated in this iteration.

Disassembly Sequence	**Disassembly Method**	**EOL Option**	**Material**	**Fitness Value**
2 3 4	0 1 1	2 0 0	0 2 2	49.16

Table 4.11 Initial disassembly sequence results.

The complete disassembly sequence generated by the SA algorithm is presented in Table 4.13 along with the corresponding coordinates, disassembly methods, EOL processing options and the material contents of the components. The table is structured as follows: the first column represents the iteration number, while the second column lists the items detected in the product. The third, fourth and fifth columns show the exact coordinates of the components. The sixth column represents the disassembly method provided by the AHP algorithm, whereas the seventh column presents the EOL processing option, also provided by the algorithm. The eighth column provides the corresponding material content for each item while the final column represents the component selected for disassembly. The algorithm terminates when there is one single item left in the EOL product. This component (component 9 in this numerical example) is then placed at the end of the sequence.

The final EOL disassembly sequence with the corresponding disassembly method, the EOL choice based on the DM preferences and the material content for each item is provided below in Table 4.14.

Here, destructive disassembly method is represented by 0 while non-destructive disassembly is denoted by 1. Similarly, 0, 1, 2, and 3 indicate the purpose of the disassembly pointing to a specific EOL processing option; reuse, recycle, storage, and proper disposal, respectively.

Component 1

	Round 1 Pairwise Matrix				Round 2 Pairwise Matrix			
	Environmental	Economic	Social		Reuse	Recycling	Storage	Proper Disposal
Environmental	1	5/1	7/1	Reuse	1	1/5	1/1	1/3
Economic	1/5	1	3/1	Recycling	5/1	1	5/1	7/1
Social	1/7	1/3	1	Storage	1/1	1/5	1	1/3
				Proper Disposal	3/1	1/7	3/1	1
Preference Vector	**72.35**	19.32	8.33	Preference Vector	9.2	**61.4**	9.2	20.2
CR	0.06			CR	0.1			

Component 2

	Round 1 Pairwise Matrix				Round 2 Pairwise Matrix			
	Environmental	Economic	Social		Reuse	Recycling	Storage	Proper Disposal
Environmental	1	1/7	1/5	Reuse	1	1/5	1/1	1/3
Economic	7/1	1	3/1	Recycling	5/1	1	5/1	7/1
Social	5/1	1/3	1	Storage	1/1	1/5	1	1/3
				Proper Disposal	3/1	1/7	3/1	1
Preference Vector	7.38	**64.34**	28.28	Preference Vector	9.2	**61.4**	9.2	20.2
CR	0.06			CR	0.1			

Component 3

	Round 1 Pairwise Matrix				Round 2 Pairwise Matrix			
	Environmental	Economic	Social		Reuse	Recycling	Storage	Proper Disposal
Environmental	1	1/3	3/1	Reuse	1	7/1	5/1	3/1
Economic	3/1	1	5/1	Recycling	1/7	1	1/3	1/1
Social	1/3	1/5	1	Storage	1/5	3/1	1	3/1
				Proper Disposal	1/3	1/1	1/3	1
Preference Vector	26.05	**63.33**	10.62	Preference Vector	**57.6**	8.59	22.36	11.43
CR	0.03			CR	0.09			

Component 4

	Round 1 Pairwise Matrix				Round 2 Pairwise Matrix			
	Environmental	Economic	Social		Reuse	Recycling	Storage	Proper Disposal
Environmental	1	1/3	3/1	Reuse	1	7/1	5/1	3/1
Economic	3/1	1	5/1	Recycling	1/7	1	1/3	1/1
Social	1/3	1/5	1	Storage	1/5	3/1	1	3/1
				Proper Disposal	1/3	1/1	1/3	1
Preference Vector	26.05	**63.33**	10.62	Preference Vector	**57.6**	8.59	22.36	11.43
CR	0.03			CR	0.09			

Component 5

	Round 1 Pairwise Matrix				Round 2 Pairwise Matrix			
	Environmental	Economic	Social		Reuse	Recycling	Storage	Proper Disposal
Environmental	1	1/1	1/3	Reuse	1	1/5	1/1	1/3
Economic	1/1	1	1/5	Recycling	5/1	1	5/1	3/1
Social	3/1	5/1	1	Storage	1/1	1/5	1	1/3
				Proper Disposal	3/1	1/3	3/1	1
Preference Vector	18.67	15.78	**65.55**	Preference Vector	9.67	**55.49**	9.67	25.16
CR	0.03			CR	**0.09**			

Component 6

	Round 1 Pairwise Matrix				Round 2 Pairwise Matrix			
	Environmental	Economic	Social		Reuse	Recycling	Storage	Proper Disposal
Environmental	1	1/3	1/1	Reuse	1	3/1	1/5	3/1
Economic	3/1	1	5/1	Recycling	1/3	1	1/5	1/1
Social	1/1	1/5	1	Storage	5/1	5/1	1	5/1
				Proper Disposal	1/3	1/1	`	1
Preference Vector	18.67	**65.55**	15.78	Preference Vector	9.2	**61.4**	9.2	20.2
CR	0.03			CR	0.06			

Component 7

	Round 1 Pairwise Matrix				Round 2 Pairwise Matrix			
	Environmental	Economic	Social		Reuse	Recycling	Storage	Proper Disposal
Environmental	1	3/1	5/1	Reuse	1	3/1	1/5	3/1
Economic	1/3	1	1/1	Recycling	1/3	1	1/5	1/1
Social	1/5	1/1	1	Storage	5/1	5/1	1	5/1
				Proper Disposal	1/3	1/1	1/5	1
Preference Vector	18.67	**65.55**	15.78	Preference Vector	9.2	**61.4**	9.2	20.2
CR	0.03			CR	0.04			

Component 8

	Round 1 Pairwise Matrix				Round 2 Pairwise Matrix			
	Environmental	Economic	Social		Reuse	Recycling	Storage	Proper Disposal
Environmental	1	1/1	1/3	Reuse	1	3/1	5/1	7/1
Economic	1/1	1	1/5	Recycling	1/3	1	1/3	1/1
Social	3/1	5/1	1	Storage	1/5	3/1	1	3/1
				Proper Disposal	1/7	1/1	1/3	1
Preference Vector	18.67	15.78	**65.6**	Preference Vector	**57.62**	11.43	22.36	8.59
CR	0.03			CR	0.09			

Component 9

	Round 1 Pairwise Matrix				Round 2 Pairwise Matrix			
	Environmental	Economic	Social		Reuse	Recycling	Storage	Proper Disposal
Environmental	1	5/1	3/1	Reuse	1	5/1	3/1	3/1
Economic	1/5	1	1/3	Recycling	1/5	1	1/5	1/3
Social	1/3	3/1	1	Storage	1/3	5/1	1	3/1
				Proper Disposal	1/3	3/1	1/3	1
Preference Vector	**63.33**	10.62	26.05	Preference Vector	**49.09**	6.7	29.13	15.07
CR	0.03			CR	0.07			

Component 10

	Round 1 Pairwise Matrix				Round 2 Pairwise Matrix			
	Environmental	Economic	Social		Reuse	Recycling	Storage	Proper Disposal
Environmental	1	3/1	1/3	Reuse	1	3/1	1/5	3/1
Economic	1/3	1	1/5	Recycling	1/3	1	1/5	1/1
Social	3/1	5/1	1	Storage	5/1	5/1	1	5/1
				Proper Disposal	1/3	1/1	1/5	1
Preference Vector	26.05	10.62	**63.3**	Preference Vector	21.88	9.38	**59.38**	9.38
CR	0.03			CR	0.06			

Table 4.12 User preference input, AHP pairwise comparison matrices, priority vectors, and consistency ratios.

Iteration	Detected Component	X-axis	Y-axis	Z-axis	Method	EOL Option	Material	Selected Component
1	1	42	80.5	110	0	1	1	1
2	2	42	80.5	110	0	1	1	2
	3	127	89	210	1	0	2	
	4	146.333	90.333	210	1	0	2	
3	3	127	89	210	1	0	2	7
	4	146.333	90.333	210	1	0	2	
	5	57.5	88	300	0	1	2	
	6	86	31	150	1	2	0	
	7	20	50	150	1	0	0	
	8	64	35	150	1	0	0	
4	3	127	89	210	1	0	2	8
	4	146.333	90.333	210	1	0	2	
	5	57.5	88	300	0	1	2	
	6	86	31	150	1	2	0	
	8	64	35	150	1	0	0	
	9	63	54	320	1	0	2	
5	3	127	89	210	1	0	2	10
	4	146.333	90.333	210	1	0	2	
	5	57.5	88	300	0	1	2	
	6	86	31	150	1	2	0	
	9	63	54	320	1	0	2	
	10	34	33	170	1	2	0	
6	3	127	89	210	1	0	2	6
	4	146.333	90.333	210	1	0	2	
	5	57.5	88	300	0	1	2	
	6	86	31	150	1	2	0	
	9	63	54	320	1	0	2	
7	3	127	89	210	1	0	2	3
	4	146.333	90.333	210	1	0	2	
	5	57.5	88	300	0	1	2	
	9	63	54	320	1	0	2	
8	4	146.333	90.333	210	1	0	2	4
	5	57.5	88	300	0	1	2	
	9	63	54	320	1	0	2	
9	5	57.5	88	300	0	1	2	5
	9	63	54	320	1	0	2	
10	9	63	54	320	1	0	2	9

Table 4.13 Disassembly sequencing results.

Disassembly Sequence	Disassembly Method	EOL Option	Material	Fitness Value
1 2 7 8 10 6 3 4 5 9	0 0 1 1 1 1 1 1 0 1	1 1 0 0 2 2 0 0 1 0	1 2 1 1 1 1 0 0 0 0	106.43

Table 4.14 Final EOL disassembly sequence

4.4 Simulated Annealing (SA) Computation Requirements and Performance Models

The first part of this research looked into utilizing Tabu Search for disassembly sequence generation. A comparison between Genetic Algorithm and Tabu search are also provided. A scatter plot for these two runs is plotted to illustrate the relationship between the two data sets (Figure 4.12). As it can also be observed from the figure, the data sets are statistically different from one another with unequal variances and significantly low correlation. Tabu Search runs are statistically less time consuming than Genetic Algorithm runs, hence providing faster solutions to the disassembly sequencing problem

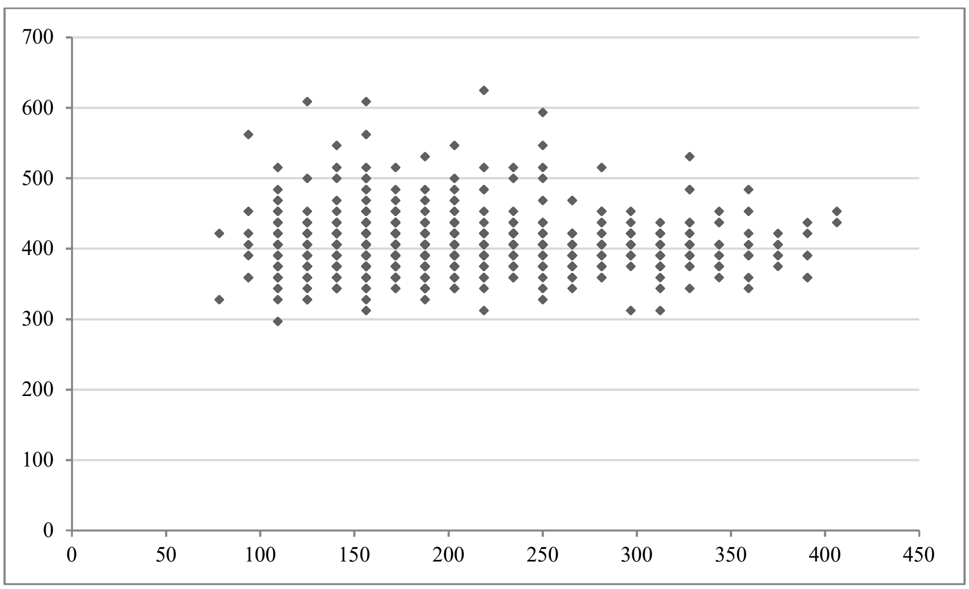


Figure 4.12 Scatter plot for Tabu search (TS) versus Genetic algorithm (GA) runs in milliseconds.

In its second step, this work also proposed a decision maker-centered disassembly sequencing algorithm. The problem is modeled as a multiple criteria decision making problem and solved via Simulated Annealing (SA) and Analytic Hierarchy Process

(AHP) methods. Utilization of SA enabled the algorithm to provide faster and reliable results while utilization of AHP introduced flexibility into the system.

Both the SA and AHP algorithms are written in C# programming language and run on The Microsoft .NET Framework 4. In order to depict the computational complexity of the SA algorithm, the simulation is run three thousand times. Figure 4.13 represents the CPU times of these runs in milliseconds.

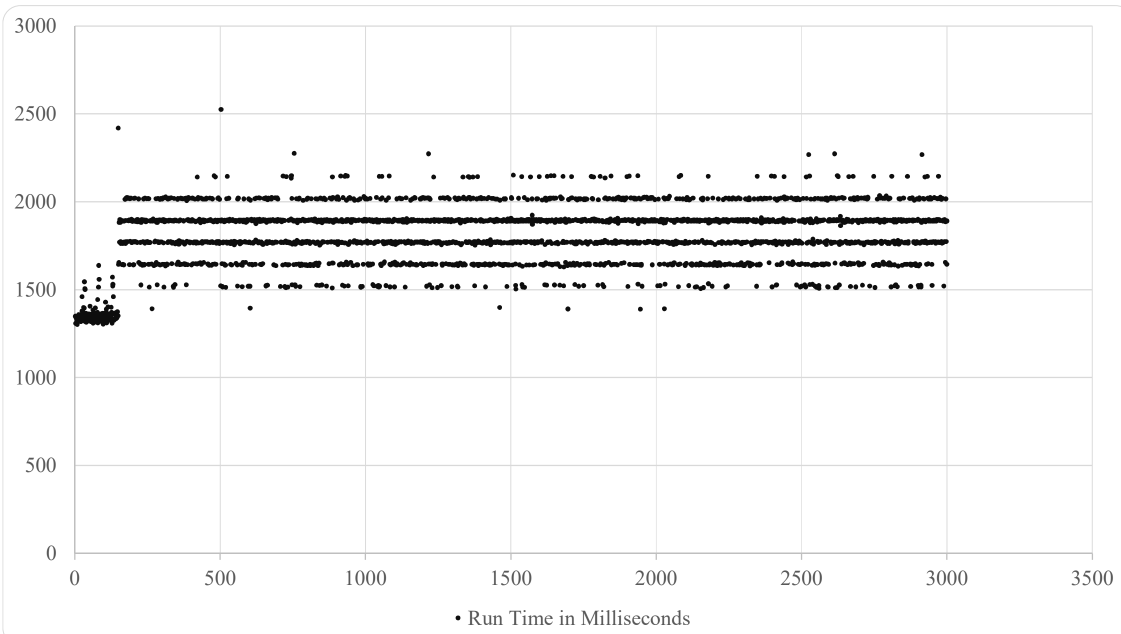


Figure 4.13 Simulated annealing run times.

The computational complexity of the Simulated Annealing algorithm is heavily reliant on the temperature annealing mechanism used for the cooling process. For instance, if the temperature is altered via a logarithmic scale, the algorithm complexity will then be higher than of an exhaustive search for the same problem with the complexity of $O(n^{n^{2n-1}})$ for an n-element search [101]. In the instances where the temperature change is proportional to the previous temperature, the algorithm complexity will be sufficiently reduced to a polynomial order $O((n^2 + n)\log(n))$ which will lead to a more efficient algorithm in large scale optimization problems [102, 103]. In this research, the temperature is proportionally altered and achieved via the Simulated Annealing for the Traveling Salesman Problem with polynomial order of complexity.

Since AHP uses the maximum eigenvalue of the pairwise matrix, the complexity of the algorithm is identical to the complexity of eigenvalue calculations. The conventional eigenvalue algorithms have order of $O(n^2)$ time complexity. In some cases, this can be further reduced to order of $O(n)$ [104, 105]. The AHP model utilized in this research calculates approximate maximum eigenvalues using one column or one row at a time. Therefore, the computational complexity of the methodology is $O(\varphi^2)$ where φ denotes the number of choices. In this study, there are 2 alternatives, (φ = 2), i.e., i. non-destructive and destructive, and, ii. Reuse and storage or recycle and disposal, resulting in $O(2^2)$.

Orthogonal arrays are used instead of full factorization 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 10 components. In addition, the robot speed and the time required for the disassembly method change 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 324 = 282,429,536,481 experiments. Using orthogonal arrays the number of experiments are reduced to 54 as shown in Table 5.1 [1].

CHAPTER 5: ROBUST DESIGN USING ORTHOGONAL ARRAYS

This section analyses the robustness of the Simulated Annealing algorithm proposed by Alshibli et al. [106] using Orthogonal Arrays (OAs) [2]. A detailed explanation of the SA implementation is also included in this section.

Figure 5.1 represent the product hierarchy and the dependencies amongst the components. The performance result in this section was compared against the results generated from an exhaustive search.

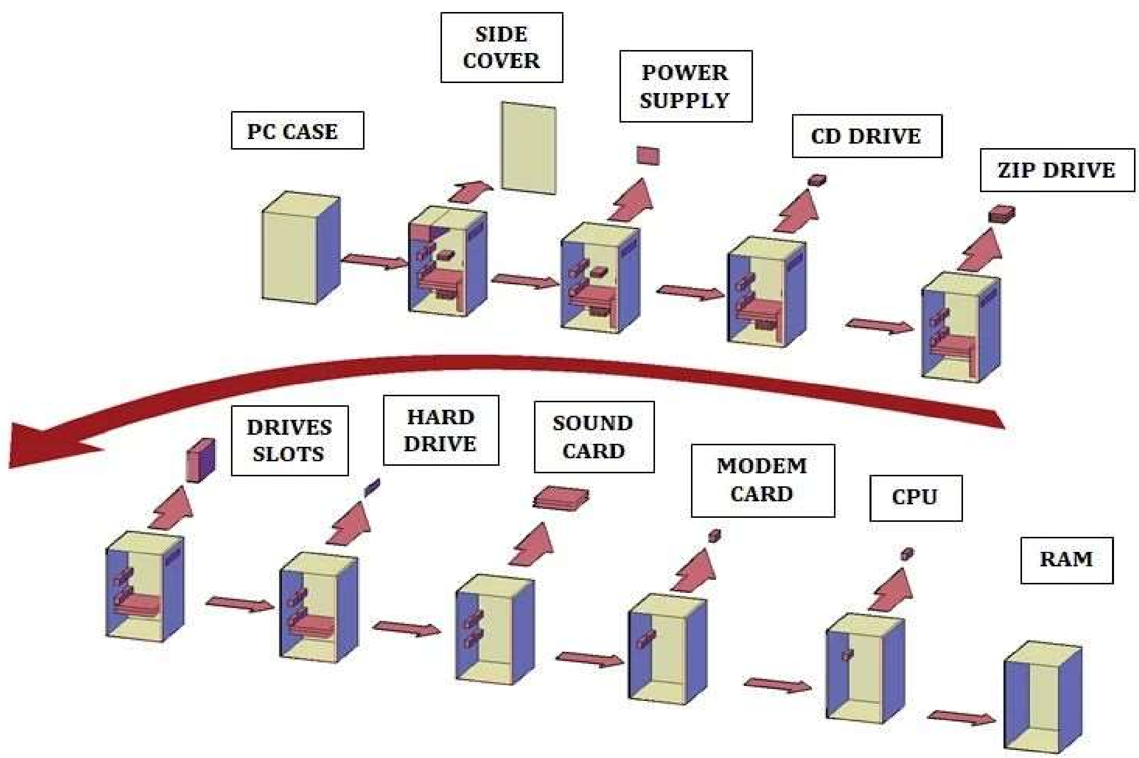


Figure 5.1 Product component hierarchy.

Expt. No.	L54 (2^1 X 3^25) Orthogonal Array Column																						
	$\mu(dt_{1j})$	$\sigma(dt_{1j})$	$\mu(dt_{2j})$	$\sigma(dt_{2j})$	$\mu(dt_{3j})$	$\sigma(dt_{3j})$	$\mu(dt_{4j})$	$\sigma(dt_{4j})$	$\mu(dt_{5j})$	$\sigma(dt_{5j})$	$\mu(dt_{6j})$	$\sigma(dt_{6j})$	$\mu(dt_{7j})$	$\sigma(dt_{7j})$	$\mu(dt_{8j})$	$\sigma(dt_{8j})$	$\mu(dt_{9j})$	$\sigma(dt_{9j})$	$\mu(dt_{10j})$	$\sigma(dt_{10j})$	$\mu(sf)$	$\sigma(sf)$	$\mu(mt_{ij})$	$\sigma(mt_{ij})$
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	1	2	2	2	2	2	2	1	1	1	1	1	1	2	3	2	3	2	3	2	3	2	3	2
5	1	2	2	2	2	2	2	2	2	2	2	2	2	3	1	3	1	3	1	3	1	3	1	3
6	1	2	2	2	2	2	2	3	3	3	3	3	3	1	2	1	2	1	2	1	2	1	2	1
7	1	3	3	3	3	3	3	1	1	1	1	1	1	3	2	3	2	3	2	3	2	3	2	3
8	1	3	3	3	3	3	3	2	2	2	2	2	2	1	3	1	3	1	3	1	3	1	3	1
9	1	3	3	3	3	3	3	3	3	3	3	3	3	2	1	2	1	2	1	2	1	2	1	2
10	2	1	1	2	2	3	3	1	1	2	2	3	3	1	1	1	1	2	3	2	3	2	3	2
11	2	1	1	2	2	3	3	2	2	3	3	1	1	2	2	2	2	3	1	3	1	3	1	3
12	2	1	1	2	2	3	3	3	3	1	1	2	2	3	3	3	3	1	2	1	2	2	1	2
13	2	2	2	3	3	1	1	1	1	2	2	3	3	2	3	2	3	3	2	3	2	1	1	1
14	2	2	2	3	3	1	1	2	2	3	3	1	1	3	1	3	1	1	3	1	3	2	2	2
15	2	2	2	3	3	1	1	3	3	1	1	2	2	1	2	1	2	2	1	2	1	3	3	3
16	2	3	3	1	1	2	2	1	1	2	2	3	3	3	2	3	2	1	1	1	1	2	3	2
17	2	3	3	1	1	2	2	2	2	3	3	1	1	1	3	1	3	2	2	2	2	3	1	3
18	2	3	3	1	1	2	2	3	3	1	1	2	2	2	1	2	1	3	3	3	3	1	2	1
19	3	1	2	1	3	2	3	1	2	1	3	2	3	1	1	2	3	1	1	3	2	2	3	3
20	3	1	2	1	3	2	3	2	3	2	1	3	1	2	2	3	1	2	2	1	3	3	1	1
21	3	1	2	1	3	2	3	3	1	3	2	1	2	3	3	1	2	3	3	2	1	1	2	2
22	3	2	3	2	1	3	1	1	2	1	3	2	3	2	3	3	2	2	3	1	1	3	2	1
23	3	2	3	2	1	3	1	2	3	2	1	3	1	3	1	1	3	3	1	2	2	1	3	2
24	3	2	3	2	1	3	1	3	1	3	2	1	2	1	2	2	1	1	2	3	3	2	1	3
25	3	3	1	3	2	1	2	1	2	1	3	2	3	3	2	1	1	3	2	2	3	1	1	2
26	3	3	1	3	2	1	2	2	3	2	1	3	1	1	3	2	2	1	3	3	1	2	2	3
27	3	3	1	3	2	1	2	3	1	3	2	1	2	2	1	3	3	2	1	1	2	3	3	1
28	1	1	3	3	2	2	1	1	3	3	2	2	1	1	1	3	2	3	2	2	3	2	3	1
29	1	1	3	3	2	2	1	2	1	1	3	3	2	2	2	1	3	1	3	3	1	3	1	2
30	1	1	3	3	2	2	1	3	2	2	1	1	3	3	3	2	1	2	1	1	2	1	2	3
31	1	2	1	1	3	3	2	1	3	3	2	2	1	2	3	1	1	1	1	3	2	3	2	2
32	1	2	1	1	3	3	2	2	1	1	3	3	2	3	1	2	2	2	2	1	3	1	3	3
33	1	2	1	1	3	3	2	3	2	2	1	1	3	1	2	3	3	3	3	2	1	2	1	1
34	1	3	2	2	1	1	3	1	3	3	2	2	1	3	2	2	3	2	3	1	1	1	1	3
35	1	3	2	2	1	1	3	2	1	1	3	3	2	1	3	3	1	3	1	2	2	2	2	1
36	1	3	2	2	1	1	3	3	2	2	1	1	3	2	1	1	2	1	2	3	3	3	3	2
37	2	1	2	3	1	3	2	1	2	3	1	3	2	1	1	2	3	3	2	1	1	3	2	2
38	2	1	2	3	1	3	2	2	3	1	2	1	3	2	2	3	1	1	3	2	2	1	3	3
39	2	1	2	3	1	3	2	3	1	2	3	2	1	3	3	1	2	2	1	3	3	2	1	1
40	2	2	3	1	2	1	3	1	2	3	1	3	2	2	3	3	2	1	1	2	3	1	1	3
41	2	2	3	1	2	1	3	2	3	1	2	1	3	3	1	1	3	2	2	3	1	2	2	1
42	2	2	3	1	2	1	3	3	1	2	3	2	1	1	2	2	1	3	3	1	2	3	3	2
43	2	3	1	2	3	2	1	1	2	3	1	3	2	3	2	1	1	2	3	3	2	2	3	1
44	2	3	1	2	3	2	1	2	3	1	2	1	3	1	3	2	2	3	1	1	3	3	1	2
45	2	3	1	2	3	2	1	3	1	2	3	2	1	2	1	3	3	1	2	2	1	1	2	3
46	3	1	3	2	3	1	2	1	3	2	3	1	2	1	1	3	2	2	3	3	2	1	1	2
47	3	1	3	2	3	1	2	2	1	3	1	2	3	2	2	1	3	3	1	1	3	2	2	3
48	3	1	3	2	3	1	2	3	2	1	2	3	1	3	3	2	1	1	2	2	1	3	3	1
49	3	2	1	3	1	2	2	1	3	2	3	1	2	2	3	1	1	3	2	1	1	2	3	3
50	3	2	1	3	1	2	3	2	1	3	1	2	3	3	1	2	2	1	3	2	2	3	1	1
51	3	2	1	3	1	2	3	3	2	1	2	3	1	1	2	3	3	2	1	3	3	1	2	2
52	3	3	2	1	2	3	1	1	3	2	3	1	2	3	2	2	3	1	1	2	3	3	2	1
53	3	3	2	1	2	3	1	2	1	3	1	2	3	1	3	3	1	2	2	3	1	1	3	2
54	3	3	2	1	2	3	1	3	2	1	2	3	1	2	1	1	2	3	3	1	2	2	1	3

Table 5.1 Reduced orthogonal array (OA), L54 ($21X3^{25}$) [1, 2].

For each parameter generated from the set of mean and standard deviation, the three levels are represented in Table 5.1 [1, 2]. The levels 1,2 and 3 are replaced by the actual values generated using the mean and standard deviation in Table 5.2.

Table 5.2 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, column 22 and 23 represent the average and standard deviation of the robot speed, whereas column 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 1,000 times using both exhaustive search and the proposed SA method. Figure 5.2 shows the results of each experiment run along with the time required to run each exhaustive search and SA model.

As it can be observed from Figure 5.2, the exhaustive search required significantly longer 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 was conducted with all disassembly sequences generated in each of the 1,000 trials to validate the results.

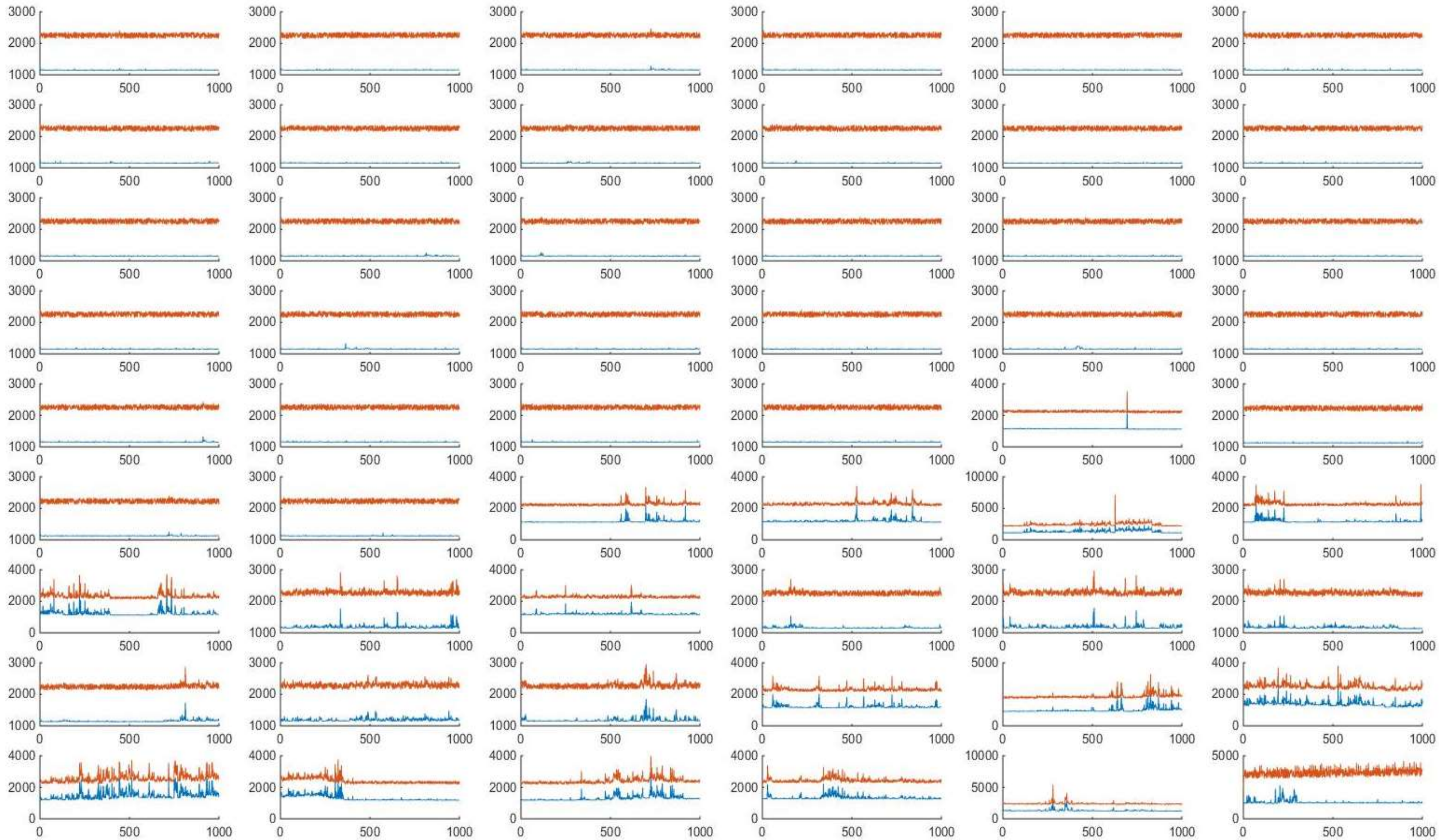


Figure 5.2 Simulated Annealing (SA) and Exhaustive Search (ES) run results.

	L54 (2^1 X 3^25) Orthogonal Array Column																							
Ex	μ(dt	σ(dt	μ(dt	σ(dt	μ(dt	σ(dt	μ(dt	σ(dt	μ(dt	σ(dt	μ(dt	σ(dt	μ(dt	σ(dt	μ(dt	σ(dt	μ(dt	σ(dt	μ(dt	σ(dt	μ(s	σ(s	μ(m	σ(m
1	2.00	0.01	3.00	0.01	3.00	0.01	2.00	0.01	3.00	0.01	4.00	0.01	2.00	0.01	1.00	0.01	3.00	0.01	2.00	0.01	7.0	0.0	0.01	1.00
2	2.00	0.01	3.00	0.01	3.00	0.01	2.00	0.01	3.25	0.05	4.25	0.05	2.25	0.01	1.25	0.05	3.25	0.05	2.25	0.05	7.2	0.0	0.05	1.25
3	2.00	0.01	3.00	0.01	3.00	0.01	2.00	0.10	3.50	0.10	4.50	0.10	2.50	0.01	1.50	0.10	3.50	0.10	2.50	0.10	7.5	0.1	0.10	1.50
4	2.00	0.05	3.25	0.05	3.25	0.05	2.25	0.01	3.00	0.01	4.00	0.01	2.00	0.01	1.50	0.05	3.50	0.05	2.50	0.05	7.5	0.0	0.10	1.25
5	2.00	0.05	3.25	0.05	3.25	0.05	2.25	0.05	3.25	0.05	4.25	0.05	2.25	0.01	1.00	0.10	3.00	0.10	2.00	0.10	7.0	0.1	0.01	1.50
6	2.00	0.05	3.25	0.05	3.25	0.05	2.25	0.10	3.50	0.10	4.50	0.10	2.50	0.00	1.25	0.01	3.25	0.01	2.25	0.01	7.2	0.0	0.05	1.00
7	2.00	0.10	3.50	0.10	3.50	0.10	2.50	0.01	3.00	0.01	4.00	0.01	2.00	0.01	1.25	0.10	3.25	0.10	2.25	0.10	7.2	0.1	0.05	1.50
8	2.00	0.10	3.50	0.10	3.50	0.10	2.50	0.05	3.25	0.05	4.25	0.05	2.25	0.00	1.50	0.01	3.50	0.01	2.50	0.01	7.5	0.0	0.10	1.00
9	2.00	0.10	3.50	0.10	3.50	0.10	2.50	0.10	3.50	0.10	4.50	0.10	2.50	0.01	1.00	0.05	3.00	0.05	2.00	0.05	7.0	0.0	0.01	1.25
10	2.25	0.01	3.00	0.05	3.25	0.10	2.50	0.01	3.00	0.05	4.25	0.10	2.50	0.00	1.00	0.01	3.00	0.05	2.50	0.05	7.5	0.0	0.10	1.25
11	2.25	0.01	3.00	0.05	3.25	0.10	2.50	0.05	3.25	0.10	4.50	0.01	2.00	0.01	1.25	0.05	3.25	0.10	2.00	0.10	7.0	0.1	0.01	1.50
12	2.25	0.01	3.00	0.05	3.25	0.10	2.50	0.10	3.50	0.01	4.00	0.05	2.25	0.01	1.50	0.10	3.50	0.01	2.25	0.01	7.2	0.0	0.01	1.25
13	2.25	0.05	3.25	0.10	3.50	0.01	2.00	0.01	3.00	0.05	4.25	0.10	2.50	0.01	1.50	0.05	3.50	0.10	2.25	0.10	7.2	0.0	0.01	1.00
14	2.25	0.05	3.25	0.10	3.50	0.01	2.00	0.05	3.25	0.10	4.50	0.01	2.00	0.01	1.00	0.10	3.00	0.01	2.50	0.01	7.5	0.0	0.05	1.25
15	2.25	0.05	3.25	0.10	3.50	0.01	2.00	0.10	3.50	0.01	4.00	0.05	2.25	0.00	1.25	0.01	3.25	0.05	2.00	0.05	7.0	0.1	0.10	1.50
16	2.25	0.10	3.50	0.01	3.00	0.05	2.25	0.01	3.00	0.05	4.25	0.10	2.50	0.01	1.25	0.10	3.25	0.01	2.00	0.01	7.0	0.0	0.10	1.25
17	2.25	0.10	3.50	0.01	3.00	0.05	2.25	0.05	3.25	0.10	4.50	0.01	2.00	0.00	1.50	0.01	3.50	0.05	2.25	0.05	7.2	0.1	0.01	1.50
18	2.25	0.10	3.50	0.01	3.00	0.05	2.25	0.10	3.50	0.01	4.00	0.05	2.25	0.01	1.00	0.05	3.00	0.10	2.50	0.10	7.5	0.0	0.05	1.00
19	2.50	0.01	3.25	0.01	3.50	0.05	2.50	0.01	3.25	0.01	4.50	0.05	2.50	0.00	1.00	0.05	3.50	0.01	2.00	0.10	7.2	0.0	0.10	1.50
20	2.50	0.01	3.25	0.01	3.50	0.05	2.50	0.05	3.50	0.05	4.00	0.10	2.00	0.01	1.25	0.10	3.00	0.05	2.25	0.01	7.5	0.1	0.01	1.00
21	2.50	0.01	3.25	0.01	3.50	0.05	2.50	0.10	3.00	0.10	4.25	0.01	2.25	0.01	1.50	0.01	3.25	0.10	2.50	0.05	7.0	0.0	0.05	1.25
22	2.50	0.05	3.50	0.05	3.00	0.10	2.00	0.01	3.25	0.01	4.50	0.05	2.50	0.01	1.50	0.10	3.25	0.05	2.50	0.01	7.0	0.1	0.05	1.00
23	2.50	0.05	3.50	0.05	3.00	0.10	2.00	0.05	3.50	0.05	4.00	0.10	2.00	0.01	1.00	0.01	3.50	0.10	2.00	0.05	7.2	0.0	0.10	1.25
24	2.50	0.05	3.50	0.05	3.00	0.10	2.00	0.10	3.00	0.10	4.25	0.01	2.25	0.00	1.25	0.05	3.00	0.01	2.25	0.10	7.5	0.0	0.01	1.50
25	2.50	0.10	3.00	0.10	3.25	0.01	2.25	0.01	3.25	0.01	4.50	0.05	2.50	0.01	1.25	0.01	3.00	0.10	2.25	0.05	7.5	0.0	0.01	1.25
26	2.50	0.10	3.00	0.10	3.25	0.01	2.25	0.05	3.50	0.05	4.00	0.10	2.00	0.00	1.50	0.05	3.25	0.01	2.50	0.10	7.0	0.0	0.05	1.50
27	2.50	0.10	3.00	0.10	3.25	0.01	2.25	0.10	3.00	0.10	4.25	0.01	2.25	0.01	1.00	0.10	3.50	0.05	2.00	0.01	7.2	0.1	0.10	1.00
28	2.00	0.01	3.50	0.10	3.25	0.05	2.00	0.01	3.50	0.10	4.25	0.05	2.00	0.00	1.00	0.10	3.25	0.10	2.25	0.05	7.5	0.0	0.10	1.00
29	2.00	0.01	3.50	0.10	3.25	0.05	2.00	0.05	3.00	0.01	4.50	0.10	2.25	0.01	1.25	0.01	3.50	0.01	2.50	0.10	7.0	0.1	0.01	1.25
30	2.00	0.01	3.50	0.10	3.25	0.05	2.00	0.10	3.25	0.05	4.00	0.01	2.50	0.01	1.50	0.05	3.00	0.05	2.00	0.01	7.2	0.0	0.05	1.50
31	2.00	0.05	3.00	0.01	3.50	0.10	2.25	0.01	3.50	0.10	4.25	0.05	2.00	0.01	1.50	0.01	3.00	0.01	2.00	0.10	7.2	0.1	0.05	1.25
32	2.00	0.05	3.00	0.01	3.50	0.10	2.25	0.05	3.00	0.01	4.50	0.10	2.25	0.01	1.00	0.05	3.25	0.05	2.25	0.01	7.5	0.0	0.10	1.50
33	2.00	0.05	3.00	0.01	3.50	0.10	2.25	0.10	3.25	0.05	4.00	0.01	2.50	0.00	1.25	0.10	3.50	0.10	2.50	0.05	7.0	0.0	0.01	1.00
34	2.00	0.10	3.25	0.05	3.00	0.01	2.50	0.01	3.50	0.10	4.25	0.05	2.00	0.01	1.25	0.05	3.50	0.05	2.50	0.01	7.0	0.0	0.01	1.50
35	2.00	0.10	3.25	0.05	3.00	0.01	2.50	0.05	3.00	0.01	4.50	0.10	2.25	0.00	1.50	0.10	3.00	0.10	2.00	0.05	7.2	0.0	0.05	1.00
36	2.00	0.10	3.25	0.05	3.00	0.01	2.50	0.10	3.25	0.05	4.00	0.01	2.50	0.01	1.00	0.01	3.25	0.01	2.25	0.10	7.5	0.1	0.10	1.25
37	2.25	0.01	3.25	0.10	3.00	0.10	2.25	0.01	3.25	0.10	4.00	0.10	2.25	0.00	1.00	0.05	3.50	0.10	2.25	0.01	7.0	0.1	0.05	1.25
38	2.25	0.01	3.25	0.10	3.00	0.10	2.25	0.05	3.50	0.01	4.25	0.01	2.50	0.01	1.25	0.10	3.00	0.01	2.50	0.05	7.2	0.0	0.10	1.50
39	2.25	0.01	3.25	0.10	3.00	0.10	2.25	0.10	3.00	0.05	4.50	0.05	2.00	0.01	1.50	0.01	3.25	0.05	2.00	0.10	7.5	0.0	0.01	1.00
40	2.25	0.05	3.50	0.01	3.25	0.01	2.50	0.01	3.25	0.10	4.00	0.10	2.25	0.01	1.50	0.10	3.25	0.01	2.00	0.05	7.5	0.0	0.01	1.50
41	2.25	0.05	3.50	0.01	3.25	0.01	2.50	0.05	3.50	0.01	4.25	0.01	2.50	0.01	1.00	0.01	3.50	0.05	2.25	0.10	7.0	0.0	0.05	1.00
42	2.25	0.05	3.50	0.01	3.25	0.01	2.50	0.10	3.00	0.05	4.50	0.05	2.00	0.00	1.25	0.05	3.00	0.10	2.50	0.01	7.2	0.1	0.10	1.25
43	2.25	0.10	3.00	0.05	3.50	0.05	2.00	0.01	3.25	0.10	4.00	0.10	2.25	0.01	1.25	0.01	3.00	0.05	2.50	0.10	7.2	0.0	0.10	1.00
44	2.25	0.10	3.00	0.05	3.50	0.05	2.00	0.05	3.50	0.01	4.25	0.01	2.50	0.00	1.50	0.05	3.25	0.10	2.00	0.01	7.5	0.1	0.01	1.25
45	2.25	0.10	3.00	0.05	3.50	0.05	2.00	0.10	3.00	0.05	4.50	0.05	2.00	0.01	1.00	0.10	3.50	0.01	2.25	0.05	7.0	0.0	0.05	1.50
46	2.50	0.01	3.50	0.05	3.50	0.01	2.25	0.01	3.50	0.05	4.50	0.01	2.25	0.00	1.00	0.10	3.25	0.05	2.50	0.10	7.2	0.0	0.01	1.25
47	2.50	0.01	3.50	0.05	3.50	0.01	2.25	0.05	3.00	0.10	4.00	0.05	2.50	0.01	1.25	0.01	3.50	0.10	2.00	0.01	7.5	0.0	0.05	1.50
48	2.50	0.01	3.50	0.05	3.50	0.01	2.25	0.10	3.25	0.01	4.25	0.10	2.00	0.01	1.50	0.05	3.00	0.01	2.25	0.05	7.0	0.1	0.10	1.00
49	2.50	0.05	3.00	0.10	3.00	0.05	2.25	0.01	3.50	0.05	4.50	0.01	2.25	0.01	1.50	0.01	3.00	0.10	2.25	0.01	7.0	0.0	0.10	1.50
50	2.50	0.05	3.00	0.10	3.00	0.05	2.50	0.05	3.00	0.10	4.00	0.05	2.50	0.01	1.00	0.05	3.25	0.01	2.50	0.05	7.2	0.1	0.01	1.00
51	2.50	0.05	3.00	0.10	3.00	0.05	2.50	0.10	3.25	0.01	4.25	0.10	2.00	0.00	1.25	0.10	3.50	0.05	2.00	0.10	7.5	0.0	0.05	1.25
52	2.50	0.10	3.25	0.01	3.25	0.10	2.00	0.01	3.50	0.05	4.50	0.01	2.25	0.01	1.25	0.05	3.50	0.01	2.00	0.05	7.5	0.1	0.05	1.00
53	2.50	0.10	3.25	0.01	3.25	0.10	2.00	0.05	3.00	0.10	4.00	0.05	2.50	0.00	1.50	0.10	3.00	0.05	2.25	0.10	7.0	0.0	0.10	1.25
54	2.50	0.10	3.25	0.01	3.25	0.10	2.00	0.10	3.25	0.01	4.25	0.10	2.00	0.01	1.00	0.01	3.25	0.10	2.50	0.01	7.2	0.0	0.01	1.50

Table 5.2 Experimental frame, modified from L54(21X3^{25})[1, 2].

CHAPTER 6: TASK ALLOCATION

This section introduces multiple robot arms to the problem environment ensuring that all robot arms work with a balanced load.

Equation 6.1 represents the part assignment status; 1 if the product is assigned and 0 if not assigned.

$$x_{jk} = \begin{cases} 1 & if\ part\ j\ is\ assigned\ to\ station\ k \\ 0 & otherwise \end{cases} \tag{6.1}$$

$$\left[x_{jk}\right]_{n\times m}, (n)\ number\ of\ parts, and\ (m)\ number\ of\ machines \tag{6.2}$$

Here, x_{ik} represents the time required to disassemble the component and the total load on the current station. In Equation 6.2, the variable n represents the number of discovered items and m represents the number of available stations or robot arms for disassembly.

$$c = \left(\sum_{i=1}^{n} dt_i\right)/n \tag{6.3}$$

In Equation 6.3, the variable c represents cycle time, viz., 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 (6.4). The main factor in this evaluation function is to minimize the number of robots running while keeping the entire system balanced. The number of stations is set to a constant value of 5.

$$\min f_1 = m \tag{6.4}$$

In Equation (6.5), main factor is balance the load on all the stations, and this is applied by calculating the square difference between the constant factor from Equation (6.3) and the total time the station is running.

$$\min f_2 = \sum_{j=1}^{m} (c - ST_j)^2 \tag{6.5}$$

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

$$\min f_3 = \sum_{i=1}^{n} i \times h_{PS_i} \quad , h_{PS_i} = \begin{cases} 1 & hazardous \\ 0 & otherwise \end{cases} \tag{6.6}$$

The final evaluation function is represented in Equation (6.7), this equation represents the demand measure This measure 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

$$\min f_4 = \sum_{i=1}^{n} i \times d_{PS_i} \quad , d_{PS_i} \in N, PS_i \tag{6.7}$$

Subject to

$$\sum_{k=1}^{m} X_{ji} = 1, j = 1, \ldots., n \tag{6.8}$$

Figure 6.1 represents the combined SA disassembly sequence generation and SA task allocation. In the initial step, the system will detect any available items for disassembly and, if there are objects detected, then SA will run and generate the disassembly sequence for the list of items. The optimum or near-optimum solution will be passed to the 3rd phase to allocate tasks and find the optimum task allocation using SA. The process will continuously execute until all the items are disassembled successfully. The result of the proposed solution is presented in Table 6.1.

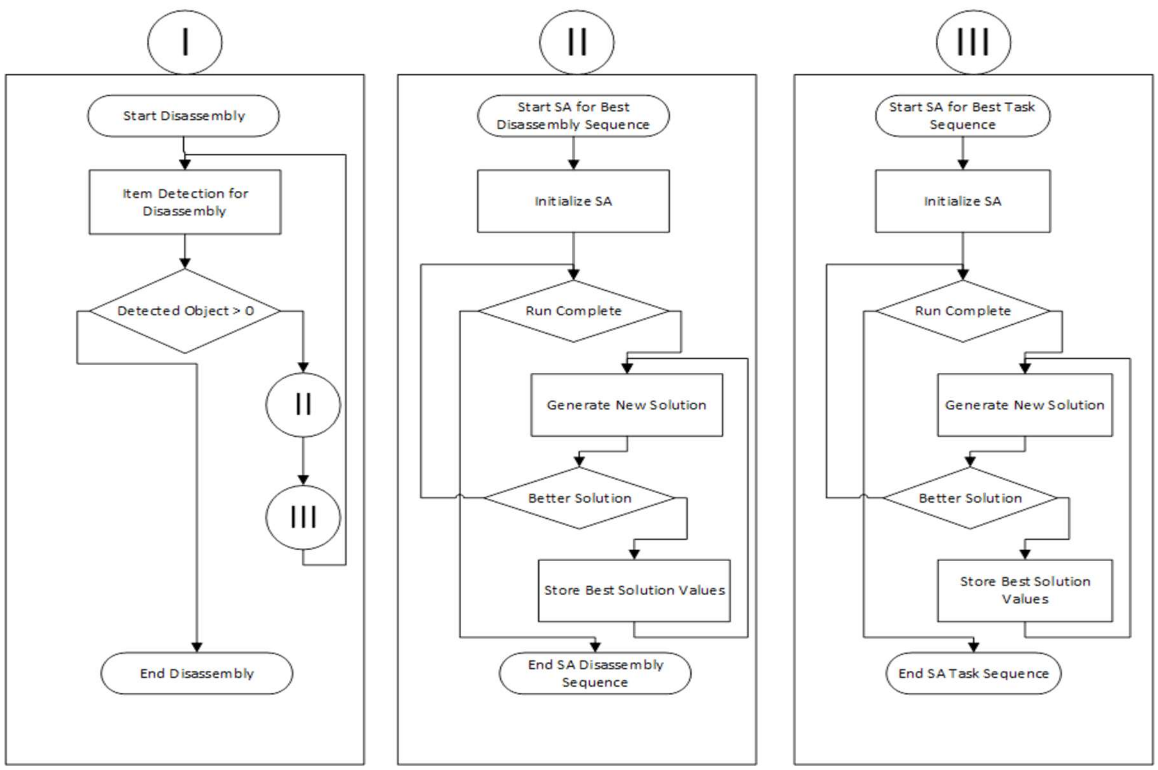


Figure 6.1 Disassembly sequence and task allocation process.

Iteration	Sequence	Allocation	Fitness Value
1	0 1 D r A 24.7381249836807	[1 0 0] [3 0 0]	1, 3, 0, 0
2	0 2 3 4 DNN ruu CPP 49.1646115780334	[1 0 0] [0 1 0] [0 0 1] [6 0 0] [0 3 0] [0 0 3]	3, 6, 0, 0
3	0 7 8 6 5 NNNN rrru AAAP 73.8853195338248	[1 0 0] [0 1 0] [0 0 1] [0 1 0] [5 0 0] [0 3 0] [0 0 4] [0 4 0]	3, 5, 0, 0
4	0 10 9 DN rs AP 53.241942485292	[1 0 0] [0 1 0] [3 0 0] [0 2 0]	2, 6, 0, 0

Table 6.1 Disassembly sequence and task allocation results.

Finally, the task allocation allows multiple robots to disassemble the product components while making sure that the robot work load remains balanced.

CHAPTER 7: CONCLUSIONS AND FUTURE RESEARCH

7.1 Conclusions

Proposed automated disassembly framework for end-of-life electronic products is able to incorporate decision makers' (DMs') preferences into the problem environment for efficient material and component recovery. The approach utilized an Analytic Hierarchy Process (AHP) model to incorporate DMs' verbal expressions into the decision problem. These results are then fed into a metaheuristic algorithm to obtain the optimum or near-optimum disassembly sequence. In this step, a metaheuristic technique, Simulated Annealing (SA) algorithm, is used. A numerical example is provided to demonstrate the functionality of the proposed approach. The disassembly sequence preserved the precedence relationships and considered the exact location of each component in the EOL product. The utilization of captured images makes the algorithm suitable for both partial and complete disassembly. That is, complete disassembly is not mandated by the simulated annealing algorithm. Furthermore, a stringency factor was included in the AHP model, to ensure overall efficiency of the disassembly operations.

It is important to note that, for small numbers of electronic products, a single arm robot can efficiently conduct disassembly operations under strict time constraints. However, when the number of EOL products rises to larger volumes, a single resource might cause bottlenecks in the disassembly lines. This issue can be addressed by introducing multiple arm robots with a load balancing system to enhance the performance of large scale disassembly processes. Furthermore, an automated system based on industrial data mining results and part testing data can be used to classify the available parts with its conditions. This would help in generating automatic scales for the decision-making algorithm and the following disassembly computations this was taken in consideration in task allocation section. Table 7.1, shows the three phases of this research.

Algorithm	**Robot Arms**	**BOM**	**Decision Making Support**
Disassembly sequencing using Tabu search [12]	Single Robot Arm	Multiple Products	No Decision-Making Support
A Decision Maker-Centered End-of-Life Product Recovery System for Robot Task Sequencing	Single Robot Arm	Multiple Products	Supports Decision-Making using AHP
Mobile Support Balanced Multi-Robots with Conscious Sequence Generation System for End-of-Life Electronic Products Disassembly	Multiple Robot Arms	Multiple Products	Support Decision-Making using AHP

Table 7.1 Research Phases

7.2 Discussion and Future Research

In any optimization problem there are two major issues which need to be considered, namely, resource utilization and execution time. Serverless architecture detects resources for the problem to be executed without interference from other processes. The architecture is supported by reputable research companies in the field of Machine Learning and AI such as: Google, Microsoft, IBM, and Amazon. This strong market acceptance gives the architecture a promising future for its expansion. Serverless architecture provides a complete tool allowing its users to monitor the processing status via a manageable dashboard. The dashboard takes into 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.

The architecture also supports a variety of programming languages, providing researchers with the flexibility to build a system using a wide range of programming languages. Currently, there are studies on creating a standard architecture to make it a more uniform and conflict-free.

With these aforementioned advantages, this research recommends using serverless architecture in solving provided problems given that the computational time would be significantly reduced.

REFERENCES

[1] M. S. Phadke, Quality engineering using robust design: Prentice Hall PTR, 1995.

[2] G. Taguchi and Y. Wu, Introduction to off-line quality control: Central Japan Quality Control Assoc., 1979.

[3] Y. Wind and T. L. Saaty, "Marketing applications of the analytic hierarchy process," Management Science, vol. 26, pp. 641-658, 1980.

[4] F. S. Hillier and G. J. Lieberman, Introduction to stochastic models in operations research: McGraw-Hill Companies, 1990.

[5] F. Hillier and G. Lieberman, "Introduction to Operations Research, McGraw Hill," New York, 2001.

[6] J. Rajgopal, "Principles and applications of operations research," Maynard's Industrial Engineering Handbook.–2004.–P, pp. 11.27-11.44, 2004.

[7] R. Ackoff and M. Sasieni, "Fundamentals of OR," ed: John Wiley & Sons, NY, 1968.

[8] D. A. Heger, "An introduction to operations research—benefits, methods and application," Austin, TX, USA: Fortuitous Technology, 2006.

[9] H. A. Taha, Operations research: an introduction: Macmillan, 1992.

[10] S. Hillier Frederick and J. Lieberman Gerald, "Introduction to operations research," ed: Boston: McGraw-Hill, 2005.

[11] F. Glover, "Tabu search-part I," ORSA Journal on Computing, vol. 1, pp. 190-206, 1989.

[12] M. Alshibli, A. El Sayed, E. Kongar, T. M. Sobh, and S. M. Gupta, "Disassembly sequencing using tabu search," Journal of Intelligent & Robotic Systems, vol. 82, pp. 69-79, 2016.

[13] E.P.A. USA Environmental Protection Agency, "2013 Report," 2013.

[14] C. Hagelüken and U. P. M. Refining, "Opportunities & challenges to recover scarce and valuable metals from electronic devices," in Vortrag Anlässlich der OECD-UNEP Conference on Resource Efficiency, Paris, 2008.

[15] Lisa Jackson, "Environmental responsibility report 2016 progress report, covering fiscal year 2015," 2016.

[16] C. P. Baldé, 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/Geneva/Vienna., November 2017.

[17] D. B. Mueller, J. Cao, E. Kongar, M. Altonji, P.-H. Weiner, and a. T. Graedel, "Service lifetimes of mineral end uses," Research Supported by the US Geological Survey, Award, 2007.

[18] S. M. McGovern and S. M. Gupta, "A balancing method and genetic algorithm for disassembly line balancing," European Journal of Operational Research, vol. 179, pp. 692-708, 2007.

[19] C. B. Kalayci and S. M. Gupta, "Simulated annealing algorithm for solving sequence-dependent disassembly line balancing problem," IFAC Proceedings Volumes, vol. 46, pp. 93-98, 2013/01/01 2013.

[20] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," Evolutionary Computation, vol. 3, pp. 1-16, 1995.

[21] L. M. Galantucci, G. Percoco, and R. Spina, "Assembly and disassembly planning by using fuzzy logic & genetic algorithms," International Journal of Advanced Robotic Systems, vol. 1, pp. 67-74, June 2004.

[22] E. Kongar and S. M. Gupta, "Disassembly sequencing using genetic algorithm," International Journal of Advanced Manufacturing Technology, vol. 30, pp. 497-506, 2006.

[23] A. ElSayed, M. Rizk, and T. Sobh, "Microcontroller implementation of fuzzy guidance," 2014.

[24] A. ElSayed, E. Kongar, and S. M. Gupta, "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.

[25] F. Torres, P. Gil, S. T. Puente, J. Pomares, and R. Aracil, "Automatic PC disassembly for component recovery," International Journal of Advanced Manufacturing Technology, vol. 23, pp. 39-46, 2004.

[26] A. ElSayed, E. Kongar, S. Gupta, and T. Sobh, "A Robotic-driven disassembly sequence generator for end-of-life electronic products," Journal of Intelligent & Robotic Systems, vol. 68, pp. 43-52, 2012/09/01 2012.

[27] B. Xing, W.-J. Gao, F. V. Nelwamondo, K. Battle, and T. Marwala, "Soft computing in product recovery: a survey focusing on remanufacturing system," ICIC Express Letters, vol. 6, pp. 23-27, 2012.

[28] C. B. Kalayci and S. M. Gupta, "Tabu search for disassembly line balancing with multiple objectives," presented at the 41st International Conference on Computers and Industrial Engineering (CIE41), 2011.

[29] F. Torres, S. Puente, and C. Díaz, "Automatic cooperative disassembly robotic system: task planner to distribute tasks among robots," Control Engineering Practice, vol. 17, pp. 112-121, 2009., vol. 17, pp. 112-121, 2009.

[30] S. Puente, F. Torres, O. Reinoso, and L. Paya, "Disassembly planning strategies for automatic material removal," The International Journal of Advanced Manufacturing Technology, vol. 46, pp. 339-350, 2010.

[31] B.-V. Kuren, "Flexible robotic demanufacturing using real time tool path generation," Robotics and Computer-Integrated Manufacturing, vol. 22, pp. 17-24, 2006.

[32] T. Saaty, "An eigenvalue allocation model for prioritization and planning, energy management and policy center, University of Pennsylvania," Working Paper1972.

[33] M. Brunelli, "Introduction to the analytic hierarchy process," 2015.

[34] E. Koç and H. A. Burhan, "An analytic hierarchy process (AHP) approach to a real world supplier selection problem: a case study of carglass turkey," Global Business and Management Research, vol. 6, p. 1, 2014.

[35] E. U. Choo and W. C. Wedley, "A common framework for deriving preference values from pairwise comparison matrices," Computers & Operations Research, vol. 31, pp. 893-908, 2004.

[36] T. L. Saaty and L. G. Vargas, "Comparison of eigenvalue, logarithmic least squares and least squares methods in estimating ratios," Mathematical Modelling, vol. 5, pp. 309-324, 1984.

[37] T. L. Saaty, "Decision making with the analytic hierarchy process," International Journal of Services Sciences, vol. 1, pp. 83-98, 2008.

[38] M. Moore, "Selecting a contractor for fast-track projects. 1. principles of contractor evaluation," Plant Engineering, vol. 39, pp. 74-75, 1985.

[39] E. Triantaphyllou and S. H. Mann, "Using the analytic hierarchy process for decision making in engineering applications: some challenges," International Journal of Industrial Engineering: Applications and Practice, vol. 2, pp. 35-44, 1995.

[40] T. Chakraborty, T. Ghosh, and P. K. Dan, "Application of analytic hierarchy process and heuristic algorithm in solving vendor selection problem," Business Intelligence Journal, vol. 4, pp. 167-177, 2011.

[41] I. Syamsuddin and J. Hwang, "The application of AHP model to guide decision makers: a case study of e-banking security," in Computer Sciences and

Convergence Information Technology, 2009. ICCIT'09. Fourth International Conference, 2009, pp. 1469-1473.

[42] D. Dalalah, F. Al-Oqla, and M. Hayajneh, "Application of the analytic hierarchy process (AHP) in multi-criteria analysis of the selection of cranes," Jordan Journal of Mechanical and Industrial Engineering, vol. 4, pp. 567-578, 2010.

[43] K. M. A.-S. Al-Harbi, "Application of the AHP in project management," International Journal of Project Management, vol. 19, pp. 19-27, 2001.

[44] E. Koç and H. A. Burhan, "An application of analytic hierarchy process (AHP) in a real world problem of store location selection," Advances in Management and Applied Economics, vol. 5, p. 41, 2015.

[45] R. Parameshwaran, S. Praveen Kumar, and K. Saravanakumar, "An integrated fuzzy MCDM based approach for robot selection considering objective and subjective criteria," Applied Soft Computing, vol. 26, pp. 31-41, 1 2015.

[46] J. Wang, M. Li, Y. Liu, H. Zhang, W. Zou, and L. Cheng, "Safety assessment of shipping routes in the South China Sea based on the fuzzy analytic hierarchy process," Safety Science, vol. 62, pp. 46-57, 2014.

[47] S. Malik, A. Kumari, and S. Agrawal, "Decision making using AHP under reverse supply chain environment for EOL products," 2015.

[48] H. Ahonen, A. G. de Alvarenga, and A. R. S. Amaral, "Simulated annealing and tabu search approaches for the corridor allocation problem," European Journal of Operational Research, vol. 232, pp. 221-233, 1/1/ 2014.

[49] C. B. Kalayci and S. M. Gupta, "A particle swarm optimization algorithm with neighborhood-based mutation for sequence-dependent disassembly line balancing problem," The International Journal of Advanced Manufacturing Technology, vol. 69, pp. 197-209, 2013.

[50] C. B. Kalayci and S. M. Gupta, "Solving sequence-dependent disassembly line balancing problem using a hybrid genetic algorithm," Proceedings for the Northeast Region Decision Sciences Institute (NEDSI), pp. 1119-1128, 2013.

[51] C. B. Kalayci, O. Polat, and S. M. Gupta, "A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem," Annals of Operations Research, vol. 242, pp. 321-354, 2016.

[52] E. Kongar and S. M. Gupta, "Goal programming approach to the remanufacturing supply-chain model," in Intelligent Systems and Smart Manufacturing, 2001, pp. 167-178.

[53] A. Z. Massoud and S. M. Gupta, "Preemptive goal programming for solving the multi-criteria disassembly-to-order problem under stochastic yields, limited supply, and quantity discount," 2010.

[54] Y. Kinoshita, T. Yamada, S. M. Gupta, A. Ishigaki, and M. Inoue, "Analysis of environmental and economic disassembly parts selection by goal programming," Procedia CIRP, vol. 40, pp. 162-167, 2016.

[55] Y. Kinoshita, T. Yamada, S. M. Gupta, A. Ishigaki, and M. Inoue, "Disassembly parts selection and analysis for recycling rate and cost by goal programming," Journal of

Advanced Mechanical Design, Systems, and Manufacturing, vol. 10, pp. JAMDSM0052-JAMDSM0052, 2016.

[56] K. Igarashi, T. Yamada, S. M. Gupta, M. Inoue, and N. Itsubo, "Disassembly system modeling and design with parts selection for cost, recycling and CO 2 saving rates using multi criteria optimization," Journal of Manufacturing Systems, vol. 38, pp. 151-164, 2016.

[57] M. K. Ghorabaee, E. K. Zavadskas, M. Amiri, and Z. Turskis, "Extended EDAS method for fuzzy multi-criteria decision-making: an application to supplier selection," International Journal of Computers Communications & Control, vol. 11, pp. 358-371, 2016.

[58] M. A. Ilgin, S. M. Gupta, and O. Battaïa, "Use of MCDM techniques in environmentally conscious manufacturing and product recovery: state of the art," Journal of Manufacturing Systems, vol. 37, Part 3, pp. 746-758, 10// 2015.

[59] C. B. Kalayci, A. Hancilar, A. Gungor, and S. M. Gupta, "Multi-objective fuzzy disassembly line balancing using a hybrid discrete artificial bee colony algorithm," Journal of Manufacturing Systems, vol. 37, Part 3, pp. 672-682, 10// 2015.

[60] C. B. Kalayci, O. Polat, and S. M. Gupta, "A variable neighbourhood search algorithm for disassembly lines," Journal of Manufacturing Technology Management, vol. 26, pp. 182-194, 2015.

[61] O. Ondemir and S. M. Gupta, "A multi-criteria decision making model for advanced repair-to-order and disassembly-to-order system," European Journal of Operational Research, vol. 233, pp. 408-419, 3/1/ 2014.

[62] O. Ondemir, M. A. Ilgin, and S. M. Gupta, "Optimal end-of-life management in closed-loop supply chains using RFID and sensors," IEEE Transactions on Industrial Informatics, vol. 8, pp. 719-728, 2012.

[63] S. Vongbunyong, S. Kara, and M. Pagnucco, "Basic behaviour control of the vision-based cognitive robotic disassembly automation," Assembly Automation, vol. 33, pp. 38-56, 2013.

[64] X. Wang and H. K. Chan, "An integrated fuzzy approach for evaluating remanufacturing alternatives of a product design," Journal of Remanufacturing, vol. 3, p. 1, 2013.

[65] A. ElSayed, E. Kongar, and S. M. Gupta, "Fuzzy linear physical programming for multiple criteria decision-making under uncertainty," International Journal of Computers Communications & Control, vol. 11, pp. 26-38, 2015.

[66] A. D. Joshi, S. M. Gupta, and T. Yamada, "Selection of supplier for end-of-life products based on the optimum profit, quality level, material sales revenue and disposal weight," Innovation and Supply Chain Management: ISCM: an International Journal for Promoting Innovation Management & Supply Chain Management, vol. 8, pp. 135-140, 2014.

[67] M. A. Ilgin and S. M. Gupta, "Environmentally conscious manufacturing and product recovery (ECMPRO): a review of the state of the art," Journal of Environmental Management, vol. 91, pp. 563-591, 2010.

[68] S. M. McGovern and S. M. Gupta, "Lexicographic goal programming and assessment tools for a combinatorial production problem," Multi-Objective Optimization in Computational Intelligence: Theory and Practice, pp. 148-184, 2008.

[69] W. Ho, "Integrated analytic hierarchy process and its applications–a literature review," European Journal of Operational Research, vol. 186, pp. 211-228, 2008.

[70] K. Rousis, K. Moustakas, S. Malamis, A. Papadopoulos, and M. Loizidou, "Multi-criteria analysis for the determination of the best WEEE management scenario in Cyprus," Waste Management, vol. 28, pp. 1941-1954, 2008.

[71] V. A. Salomon, F. A. Marins, and M. A. Duduch, "Multiple-decisions decision-making applied to the supplier selection for assembly line equipments in an automotive industry," in International Symposium on the Analytic Hierarchy Process, 2007.

[72] N. Alagumurthi, K. Palaniradja, and V. Soundararajan, "Optimization of grinding process through design of experiment (DOE)—A comparative study," Materials and manufacturing processes, vol. 21, pp. 19-21, 2006.

[73] S. P. Kondapalli, S. R. Chalamalasetti, and N. R. Damera, "Application of Taguchi based design of experiments to fusion arc weld processes: A review," International Journal of Business Research and Development, vol. 4, 2013.

[74] H. K. Aksoy and S. M. Gupta, "Buffer allocation plan for a remanufacturing cell," Computers & Industrial Engineering, vol. 48, pp. 657-677, 2005.

[75] M. A. Ilgin and G. T. Taşoğlu, "Simultaneous Determination of Disassembly Sequence and Disassembly-to-Order Decisions Using Simulation Optimization," Journal of Manufacturing Science and Engineering, vol. 138, p. 101012, 2016.

[76] Y.-P. Chang, "Optimal harmonic filters design of the Taiwan high speed rail traction system of distributer generation system with specially connected transformers," International Journal of Electrical Power & Energy Systems, vol. 62, pp. 80-89, 2014.

[77] M. A. Ilgin, S. M. Gupta, and K. Nakashima, "Coping with disassembly yield uncertainty in remanufacturing using sensor embedded products," Journal of Remanufacturing, vol. 1, p. 7, 2011.

[78] L. Lazic and N. Mastorakis, "Orthogonal Array application for optimal combination of software defect detection techniques choices," WSEAS Transactions on Computers, vol. 7, pp. 1319-1336, 2008.

[79] M. A. Moghaddam and F. Kolahan, "Application of orthogonal array technique and particle swarm optimization approach in surface roughness modification when face milling AISI1045 steel parts," Journal of Industrial Engineering International, vol. 12, pp. 199-209, 2016.

[80] A. Gungor, S. M. Gupta, K. Pochampally, and S. V. Kamarthi, "Complications in disassembly line balancing," in Environmentally conscious manufacturing, 2001, pp. 289-299.

[81] S. Mete, Z. A. Çil, E. Özceylan, and K. Ağpak, "Resource constrained disassembly line balancing problem," IFAC-PapersOnLine, vol. 49, pp. 921-925, 2016.

[82] M. L. Bentaha, O. Battaïa, and A. Dolgui, "Disassembly line balancing and sequencing under uncertainty," Procedia CIRP, vol. 15, pp. 239-244, 2014.

[83] A. Gungor and S. M. Gupta, "A solution approach to the disassembly line balancing problem in the presence of task failures," International journal of production research, vol. 39, pp. 1427-1467, 2001.

[84] F. Torres, P. Gil, S. Puente, J. Pomares, and R. Aracil, "Automatic PC disassembly for component recovery," The international journal of advanced manufacturing technology, vol. 23, pp. 39-46, 2004.

[85] A. L. Gutjahr and G. L. Nemhauser, "An algorithm for the line balancing problem," Management Science, vol. 11, pp. 308-315, 1964.

[86] E. Erel and H. Gokcen, "Shortest-route formulation of mixed-model assembly line balancing problem," European Journal of Operational Research, vol. 116, pp. 194-204, 1999.

[87] P. R. McMullen and P. Tarasewich, "Using ant techniques to solve the assembly line balancing problem," Iie Transactions, vol. 35, pp. 605-617, 2003.

[88] L. Duta, F. G. Filip, and J.-M. Henrioud, "Applying equal piles approach to disassembly line balancing problem," IFAC Proceedings Volumes, vol. 38, pp. 152-157, 2005.

[89] L. Duta, F. G. Filip, and J.-M. Henrioud, "A method for dealing with multi-objective optimization problem of disassembly processes," in Assembly and Task Planning, 2003. Proceedings of the IEEE International Symposium on, 2003, pp. 163-168.

[90] A. ElSayed, E. Kongar, and S. M. Gupta, "A genetic algorithm approach to end-of-life disassembly sequencing for robotic disassembly," 2010.

[91] S. D. Morgan and R. J. Gagnon, "A systematic literature review of remanufacturing scheduling," International Journal of Production Research, vol. 51, pp. 4853-4879, 2013.

[92] S. Avikal, P. Mishra, R. Jain, and H. Yadav, "A PROMETHEE method based heuristic for disassembly line balancing problem," Industrial Engineering & Management Systems, vol. 12, pp. 254-263, 2013.

[93] G. McGrath and P. R. Brenner, "Serverless computing: Design, implementation, and performance," in Distributed Computing Systems Workshops (ICDCSW), 2017 IEEE 37th International Conference on, 2017, pp. 405-410.

[94] T. K. Ho and J. Y. Lee, "A row-permutated data reorganization algorithm for growing server-less video-on-demand systems," in Cluster Computing and the Grid, 2003. Proceedings. CCGrid 2003. 3rd IEEE/ACM International Symposium on, 2003, pp. 44-51.

[95] W. T. Leung and J. Y. Lee, "A server-less architecture for building scalable, reliable, and cost-effective video-on-demand systems," Chinese University of Hong Kong, 2002.

[96] A. Adya, W. J. Bolosky, G. Cermak, J. R. Douceur, M. M. Theimer, and R. P. Wattenhofer, "Serverless distributed file system," ed: Google Patents, 2006.

[97] W. J. Bolosky, J. R. Douceur, D. Ely, and M. Theimer, "Feasibility of a serverless distributed file system deployed on an existing set of desktop PCs," in ACM SIGMETRICS Performance Evaluation Review, 2000, pp. 34-43.

[98] S. Hendrickson, S. Sturdevant, T. Harter, V. Venkataramani, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "Serverless computation with openlambda," Elastic, vol. 60, p. 80, 2016.

[99] N. Bila, P. Dettori, A. Kanso, Y. Watanabe, and A. Youssef, "Leveraging the serverless architecture for securing linux containers," in Distributed Computing Systems Workshops (ICDCSW), 2017 IEEE 37th International Conference on, 2017, pp. 401-404.

[100] I. Baldini, P. Cheng, S. J. Fink, N. Mitchell, V. Muthusamy, R. Rabbah, et al., "The serverless trilemma: function composition for serverless computing," presented at the Proceedings of the 2017 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, Vancouver, BC, Canada, 2017.

[101] X.-S. Yang, Nature-inspired metaheuristic algorithms: Luniver Press, 2010.

[102] S. Rajasekaran, "On the convergence time of simulated annealing," Research Report MS-CIS-90-89 University of Pennsylvania, Department of Computer and Information Science, USA, November 1990., 1990.

[103] P. B. Hansen, "Simulated annealing " Electrical Engineering and Computer Science Technical Reports, p. 170, June 1992 June 1992

[104] J. Lee, V. Balakrishnan, C.-K. Koh, and D. Jiao, "A linear-time eigenvalue solver for finite-element-based analysis of large-scale wave propagation problems in on-chip interconnect structures," in Antennas and Propagation Society International Symposium, 2008. AP-S 2008. IEEE, 2008, pp. 1-4.

[105] J. Lee, V. Balakrishnan, C.-K. Koh, and D. Jiao, "From O(k2N) to O(N): A fast complex-valued eigenvalue solver for large-scale on-chip interconnect analysis," in Microwave Symposium Digest, 2009. MTT'09. IEEE MTT-S International, 2009, pp. 181-184.

[106] M. Alshibli, A. El Sayed, O. Tozanli, E. Kongar, T. M. Sobh, and S. M. Gupta, "A Decision Maker-Centered End-of-Life Product Recovery System for Robot Task Sequencing," Journal of Intelligent & Robotic Systems, pp. 1-14, 2017.