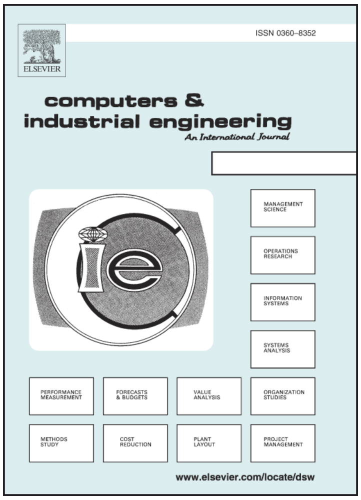
Economic modelling of robotic disassembly in end-of-life product recovery for remanufacturing

Ramirez, F. Javier; Aledo, Juan A.; Gamez, Jose A.; Pham, Duc

DOI:
10.1016/j.cie.2020.106339

Citation for published version (Harvard):
Ramirez, FJ, Aledo, JA, Gamez, JA & Pham, D 2020, 'Economic modelling of robotic disassembly in end-of-life product recovery for remanufacturing', *Computers & Industrial Engineering*, vol. 142, 106339, pp. 1-19. https://doi.org/10.1016/j.cie.2020.106339



Economic modelling of robotic disassembly in end-of-life product recovery for remanufacturing

F. Javier Ramírez, Juan A. Aledo, Jose A. Gamez, Duc T. Pham

PII: S0360-8352(20)30073-5
DOI: https://doi.org/10.1016/j.cie.2020.106339
Reference: CAIE 106339

To appear in: *Computers & Industrial Engineering*

Received Date: 5 June 2019
Revised Date: 8 January 2020
Accepted Date: 31 January 2020

Please cite this article as: Ramírez, F.J., Aledo, J.A., Gamez, J.A., Pham, D.T., Economic modelling of robotic disassembly in end-of-life product recovery for remanufacturing, *Computers & Industrial Engineering* (2020), doi: https://doi.org/10.1016/j.cie.2020.106339

Economic modelling of robotic disassembly in end-of-life product recovery for remanufacturing

F. Javier Ramírez[a,*], Juan A. Aledo[b], Jose A. Gamez[c], Duc T. Pham[d]

[a]*School of Industrial Engineering, Department of Business Administration, Universidad de Castilla-La Mancha, 02071 Albacete, Spain. Email: franciscoj.ramirez@uclm.es*

[b]*Department of Mathematics, Universidad de Castilla-La Mancha, 02071 Albacete, Spain. Email: JuanAngel.Aledo@uclm.es*

[c]*Department of Computer Systems, Universidad de Castilla-La Mancha, 02071 Albacete, Spain. Email: Jose.Gamez@uclm.es*

[d]*Department of Mechanical Engineering, College of Engineering and Physical Sciences, The University of Birmingham, B15 2TT, UK. Email: D.T.Pham@bham.ac.uk*

[*]Corresponding author

Acknowledgements

This work was partially supported by the Spanish "Ministry of the Economy and Competitiveness" and the European Union (FEDER Funds), ECO2016-75781-P project; and the Engineering and Physical Sciences Research Council (EPSRC), UK, grant no. EP/N018524/1. The authors would also like to thank the University of Birmingham (UK) and are grateful to the University of Castilla-La Mancha for funding the research stay of Dr F. Javier Ramírez at the College of Engineering and Physical Sciences, University of Birmingham, UK, in 2016 and 2017.

Economic modelling of robotic disassembly in end-of-life product recovery for remanufacturing

Abstract

The key to fully achieving the benefits of remanufacturing lies in the efficient and cost-effective reuse of components from end-of-life (EoL) products. Unlike recycling and disposal, remanufacturing contributes to a firm's profits and reduces environmental impacts. Maximising the economic value of recovery options while meeting environmental regulations is of prime importance. This paper proposes a novel model to design the robotic disassembly process (RDP) for EoL products. An optimisation decision-making model has been designed to find the near-optimal solution that achieves the best economic performance of the process while simultaneously yielding the optimal disassembly level, or "stopping point", disassembly sequence plan, and recovery option for the components in a framework. Furthermore, the model helps the robotic cell to re-plan the ongoing disassembly process. To do so, it recalculates the economic outputs, making decisions about the re-planned optimal disassembly level and the reassigned recovery option for the components: reuse, remanufacturing, recycling or disposal. The model was been tested using a case study based on a gear pump. The results demonstrate the effectiveness of the proposed model and provide insights into recovery practices for remanufacturing.

Keywords: End-of-life, Remanufacturing, Robotic disassembly, Partial disassembly, Disassembly sequence planning, Recovery option

1. Introduction

The increasing number of end-of-life (EoL) products is a major concern for governments around the world due to the enormous quantity of waste materials being generated each year. Moreover, population growth, together with the reduction in product lifetime due mainly to consumer' behaviour, will lead to a significant increase in products to be disposed of in the next decade (UNEP, 2017). Therefore, effective solutions to counter this trend and more research on ways of recovering EoL products and their components are required.

Remanufacturing is "the process of returning a used product to at least its Original Equipment Manufacturer's (OEM) performance specification from the customers' perspective, and giving the resultant product a warranty that is at least equal to that of a newly manufactured equivalent" (Matsumoto and Ijomah, 2013). Remanufacturing has significant implications for environmental preservation and firms' profits through the use of recovered components to be assembled into remanufactured products, saving on raw materials, manufacturing costs and energy consumption, in addition to reducing environmental impacts.

Because components to be remanufactured must be previously disassembled, disassembly is a critical and unavoidable step in the recovery of EoL products. Disassembly Sequence Planning (DSP) is the part of the disassembly process focused on designing a detailed disassembly plan for removing specific components or sub-assemblies from a whole product or assembly (Lambert, 2003; Zhou et al., 2018). Due to its high associated costs, disassembly has emerged as a key issue in determining the success of remanufacturing as a recovery option. Owing to its complexity, disassembly is usually conducted by humans. However, there is currently growing interest in robotic disassembly on account of its potentially higher efficiency and lower costs. In the literature, few studies have addressed the DSP problem in the framework of robotic disassembly for remanufacturing, mainly because of the high degree of uncertainty in both the disassembly process and in the condition of the returned products to be disassembled (Kalayci and Gupta, 2013; Liu et al., 2018). This gap in the literature opens up an opportunity for research.

Furthermore, partial disassembly is a growing focus of attention compared to traditional research centred on the complete disassembly process (Rickli and Camelio, 2013; Smith et al., 2016). Complete disassembly is expensive and, in most cases, unnecessary. Thus, another key issue to address is that of identifying the optimal disassembly level that determines the highest profitability of the process, and the optimal sequence to reach that threshold.

Mathematical programming methods are unsuitable for solving the problem of disassembly sequence planning as it is a Non-Deterministic Polynomial (NP) complete problem (Ghoreishi et al., 2013). Moreover, the difficulty of calculation increases with the number of components and parts in the product being disassembled. This type of problem lends itself to the application of intelligent optimisation algorithms and metaheuristic methods, as they can find near optimal solutions through the use of a reasonable number of resources (e.g. CPU time). To approach the disassembly problem by using metaheuristic algorithms, we need to provide two main components: a computational representation for potential solutions and an evaluation function to score their efficiency. By using these two components, the metaheuristic algorithm will make an intelligent randomized exploration of the search space guided by the scores returned by the evaluation function. In the problem at hand, potential search spaces are the space of permutations, where metaheuristics has been successfully applied to many problems (e.g. travelling salesman problem) or has been used to define more complex structures in order to accommodate other information features, such as the destination of each disassembled component. In addition, the evaluation function used in this case is simply the probability multi-objective function defined to compute the profit obtained in each disassembly sequence. As proof of the suitability of this family of algorithms for the disassembly problem, we can find different approaches in the specific literature, such as the Genetic Algorithm (GA) (Go et al., 2012; Gonçalves et al., 2005; Ren et al., 2018), Ant Colony Optimisation (ACO) (Dorigo et al., 2006), Particle Swarm Optimisation (PSO) (Poli et al., 2007), Artificial Bee Colony (ABC) (Karaboga, 2005), Bees Algorithm (BA) (Haj Darwish et al., 2018; Liu et al., 2018; Pham and Castellani, 2015; Pham et al., 2005, 2006), multi-objective optimisation decision-making models (Aydemir-Karadag and Turkbey, 2013; Gunantara, 2018; Meng et al., 2017; Ondemir and Gupta, 2014; Rickli and Camelio, 2013; Zhang et al., 2017), or the integer programming model (Yu and Lee, 2018).

Considering the above, research on robotic disassembly must focus on how to select the optimal disassembly level, while designing the most appropriate disassembly sequence plan, with a decision-making process for component recovery options. The purpose of this paper is to fill the gap in the literature while also determining: (1) the optimal disassembly level, (2) the disassembly sequence plan, (3) the recovery option for the components of the product to be disassembled, and (4) the re-planning of the optimal disassembly level. A decision-making model has been designed to select the optimal solution to achieve the best economic profit from the robotic disassembly process, identifying both the “stopping point”, which defines the most appropriate disassembly level, and the recovery option (reuse, remanufacturing, recycling or disposal) for the components. Furthermore, the model is able to re-plan the disassembly process using the information on the previously completed stages. Thus, the disassembly level in a partial disassembly strategy can be modified according to the process uncertainties and the condition of the product (or its components) to be disassembled.

The rest of the paper is organised as follows. Section 2 reviews the related literature that defines the background to our work. Section 3 describes the model and the associated formulation. Section 4 proposes the resolution algorithms to test the model. Section 5 is devoted to presenting the experimental case, the results of which are presented and discussed in Section 6. Finally, Section 7 concludes the paper and proposes future research.

2. Literature Review

In the literature, Disassembly Planning is defined as a Non-deterministic Polynomial (NP) complete problem (Ghoreishi et al., 2013) covering three main sub-problems: *Disassembly Sequence Planning* (DSP), *Disassembly Line Balancing* (DLB) and *Disassembly Path Planning* (DPP) (Ghandi and Masehian, 2015). In this research work, we focus on the DSP problem, which involves searching for all possible disassembly sequences and the selection of the optimal solution from among these. Numerous studies on the DSP problem have been published in recent decades. The works by Lambert (2003) and Zhou et al. (2018) include extensive surveys on disassembly sequencing drawing on the foundations of disassembly theory, considering the component-oriented approach, product-component approach and the hierarchical tree approach. There are also valuable research works considering the two major disassembly modes: complete disassembly (Gupta et al., 2004; Kongar and Gupta, 2006; Li et al., 2002) and partial disassembly (Rickli and Camelio, 2013, 2014; Smith et al., 2016; Zhang et al., 2007).

Complete or total disassembly is impractical and expensive in most cases, and unnecessary in many others (Smith et al., 2016). Thus, interest in partial/selective disassembly has recently increased, giving rise to an extensive body of literature (Zhou et al., 2018). Feldmann et al. (2001) was the first to define the optimal disassembly sequence as the balance between 'no disassembly' and 'complete disassembly', suggesting the disassembly process should be stopped at a level referred to as "stopping point". The work by Smith et al. (2016) performs a cost-benefit and rule-based analysis to find the optimised disassembly sequence based on this approach. The study by Rickli and Camelio (2013) proposes a trade-off between profitability and environmental impact in a partial disassembly sequence strategy, and Rickli and Camelio (2014) address the impact of EoL product quality uncertainty on partial disassembly sequences. Multi-objective techniques are applied by Percoco and Diella (2013) in partial disassembly planning. Furthermore, destructive operations are also considered in selective disassembly sequence planning by Wang et al. (2017).

Traditionally, the disassembly process has been managed manually, due mainly to its complexity and the high upfront costs of the machinery and facilities needed. However, robotic and autonomous disassembly has recently become the subject of greater attention because of its high efficiency and lower disassembly time and costs for large series. Robotic disassembly can also be used to replace human labour in difficult and hazardous tasks. The first significant efforts in automatic disassembly were the studies by Hesselbach (1994) who recovered valuable materials from printed circuit boards in an automatic cell, and Büker et al. (2001), who used machine vision to automatically remove wheels from used cars. Later research in robotic disassembly includes the works by Gil et al. (2007), defining a robotic system using several robots in a parallel and a cooperative way, and Torres et al. (2009), designing an automatic cooperative disassembly robotic system using a task planner based on decision trees. Furthermore, the works by Elsayed et al. (2010), using a Genetic Algorithm (GA) for disassembly sequencing of EoL products in robotic disassembly, and ElSayed et al. (2011, 2012) proposing an intelligent automated disassembly cell for online (real time) selective disassembly, were considered significant contributions to the automatic disassembly literature. More recent works by Vongbunyong et al. (2012, 2013, 2015) represent additional achievements in robotic disassembly, mainly in the use of cognitive robotics and vision-based systems focused on the reduction of the uncertainties and the variations existing in the automatic disassembly process. It is also worth highlighting the work by Barwood et al. (2015), which adopts key features in reconfigurable systems to increase flexibility and automation in recycling activities, using an especially designed robotic cell for the disassembly of electrical car components. These and other efforts have focused on the reduction of the existing uncertainties in the disassembly process, mainly originated by the condition of the returned products to be disassembled.

Decision-making models have been widely used in the optimisation of disassembly processes for remanufacturing. In the literature, disassembly process planning is assumed to involve three decision variables, as follows: (a) *Disassembly level*: decision on whether more disassembly operations are performed or not at each stage of disassembling a product (b) *Disassembly sequence*: the design of the disassembly sequence operations (c) *EoL recovery option*: decision on how each component or sub-assembly is to be dealt with (e.g., reuse, remanufacturing, recycling, disposal, etc.) (Lambert, 2003; Ma et al., 2011; Santochi et al., 2002). Economics is the main driver in most of the extensive research on decision-making models for disassembly. The work by Johnson and Wang (1998) was one of the first to take into account economic factors in the scheduling of disassembly operations for Material Recovery Opportunities (MRO), while Jovane et al. (1998) defined a framework for solving the computer-aided disassembly planning problem by the maximisation of a profit rate function of time. More recent are the studies by Ilgin and Gupta (2010), analysing the economic benefits of sensor embedded products and conventional products in a multi-product disassembly line; Rickli and Camelio (2014), evaluating disassembly sequences based on profit standard deviation and profit probability as well as the traditionally used expected profit; and Song et al. (2014), proposing a disassembly sequence planning method to reduce additional efforts of removing extra parts in selectable disassembly using a disassembly cost criteria. In addition to economic factors, the work by Meng et al. (2017) includes environmental and social criteria to address the disassembly problem from the three dimensions of sustainability, proposing multi-objective optimisation decision-making of quality dependent product recovery for sustainability, involving the selection between end-of-life product remanufacturing and dismantling.

All of these research works provide significant insights into decision-making models for disassembly. However, most consider resolving the disassembly sequence and deciding on the recovery option for the components as separate issues. Furthermore, none addresses the problem within a robotic disassembly framework, using a partial sequence strategy with re-planning capability. This is the gap in the literature that the present study aims to fill: an optimisation model to make efficient decisions in partial disassembly processes strategies with regard to process and product uncertainties.

3. Model description and formulation

3.1. Model approach

The economic making-decision model proposed in this paper consists of five stages, as shown in Figure 1.

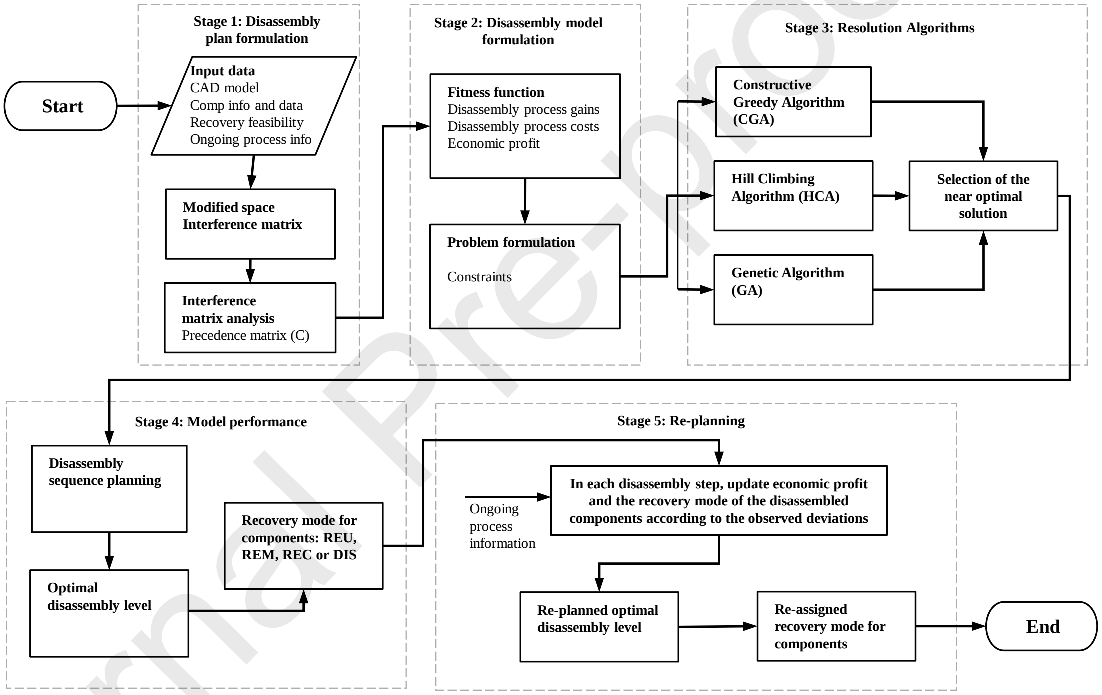

Stage 1: Disassembly plan formulation
Start
Input data
CAD model
Comp info and data
Recovery feasibility
Ongoing process info
Modified space
Interference matrix
Interference matrix analysis
Precedence matrix (C)
Stage 2: Disassembly model formulation
Fitness function
Disassembly process gains
Disassembly process costs
Economic profit
Problem formulation
Constraints
Stage 3: Resolution Algorithms
Constructive Greedy Algorithm (CGA)
Hill Climbing Algorithm (HCA)
Genetic Algorithm (GA)
Selection of the near optimal solution
Stage 4: Model performance
Disassembly sequence planning
Optimal disassembly level
Recovery mode for components: REU, REM, REC or DIS
Stage 5: Re-planning
Ongoing process information
In each disassembly step, update economic profit and the recovery mode of the disassembled components according to the observed deviations
Re-planned optimal disassembly level
Re-assigned recovery mode for components
End

Figure 1: Overview of the five-stage optimisation decision-making model

At the start of the process, complete information about the product and its components is provided to the model. CAD design is deployed to better inform the model about the product configuration and in order to define the disassembly plan formulation. Furthermore, data concerning the components is required to complete the following steps. In addition, the model permits a first selection of the final use of the components according to the user's preference for recovering certain parts of the product, to be reused, remanufactured or recycled, or whether the user intends' to dispose of some of the parts.

The first stage of the model focuses on the definition of the disassembly plan formulation, where the space interference matrix along the six directions (X+, X-, Y+, Y-, Z+, Z-) and the interference matrix to generate the feasible disassembly process are proposed (Jin et al., 2015, 2013). From the information provided by the interference analysis of the product to be disassembled, the precedence matrix $C = (c_{i,j})$ is defined as an $n\ x\ n$ matrix such that $c_{i,j} = 1$ if item i must be disassembled before item j, and $c_{i,j} = 0$ if this is not necessary. The second stage is devoted to defining the disassembly problem to be optimised. In this step, the near optimal solution (OS) based on a cost-benefit analysis is proposed. The third stage of the model proposes the resolution of the robotic disassembly optimisation problem by means of the selected algorithms. The fourth stage is devoted to the performance results. Following this, a sensitivity analysis of the main variables of the process is performed in order to determine how the evolution of key parameters impact on the economic performance of the process. Finally, the fifth step performs the re-planning of the optimal disassembly level (stopping point) of the process. It is well supported by the information from the ongoing process. If differences emerge between the performance of the ongoing process compared to the model solution, or the real condition of the disassembled components differs from the solution proposed by the model, the economic outcomes of the process could be affected. In consequence, the optimal disassembly level may be modified in relation to the initial solution. In addition, the recovery mode of certain components could be reassigned.

Once the model is applied, the disassembly process outcomes are obtained. First, the model provides the optimal disassembly level (stopping point). From this, an initial solution to the disassembly sequence planning is obtained. Based on this DSP, and on the input data, the model identifies the components to be reused (REU), remanufactured (REM), recycled (REC) or disposed of (DIS). Additionally, the solution provides the main optimised process parameters such as process time, process costs and economic profit. Finally, the model is also able to re-plan the optimal disassembly process level according to the product and process uncertainties, and depending on the stages that have been not yet completed. In this way, the optimal disassembly process level and the decision concerning the components that have not been disassembled can be modified in line with the updated information.

3.2. Model formulation

The assessment of the economic profit *(PR)* can be expressed as follows, in Eq. 1,

$$PR = DS_G - DS_C \tag{1}$$

where DS_G refers to the total disassembly process gains and DS_C expresses the total disassembly process costs.

3.2.1. Disassembly process gains

The disassembly process gains are expressed as:

$$\begin{aligned} DS_G = \sum_{i=1}^{n}\sum_{m=1}^{2} RP_i\, r_{i,m}\, \alpha_i + \sum_{i=1}^{n} RR_i\, r_{i,3}\, \alpha_i - \\ - \sum_{i=1}^{n} CD_i\, r_{i,4}\, \alpha_i - \sum_{i=1}^{n} CD_i\, r_{i,4}\, (1-\alpha_i) = \\ = \sum_{i=1}^{n}\sum_{m=1}^{2} RP_i\, r_{i,m}\, \alpha_i + \sum_{i=1}^{n} RR_i\, r_{i,3}\, \alpha_i - \sum_{i=1}^{n} CD_i\, r_{i,4} \end{aligned} \tag{2}$$

where:

- i is the indicator of each component of the product to be disassembled and varies from 1 to n.
- m is the indicator of the recovery mode:
  - 1 if component i is assigned to be reused.
  - 2 if component i is assigned to remanufactured.
  - 3 if component i is assigned to be recycled.
  - 4 if component i is assigned to be disposed of.
- RP_i is the retail price, measured as the income obtained from component i being reused or remanufactured for a new product.
- $r_{i,m}$ is an indicator of the recovery mode: 1 if mode m is assigned to component i.
- α_i is an indicator that takes the value of 1 if the component i is disassembled, and 0 otherwise.
- RR_i is the revenue obtained from component i being recycled
- CD_i is the disposal cost of component i.

3.2.2. Disassembly process costs

The disassembly process costs are obtained as follows:

$$DS_C = C_{OP} + C_{RC} + C_{OH} + C_{DP} \tag{3}$$

where:

- C_{OP} is the total operation costs of the disassembly process.
- C_{RC} is the total recovery costs of the components that will be reused, remanufactured or disposed of.
- C_{OH} is the total overhead costs of the company, and is assigned to all the components to be disassembled in the overall disassembly process.

- C_{DP} is the total depreciation costs of the machinery, and is assigned to all the components to be disassembled in the overall disassembly process.

Next, the concept and equations used to obtain the different costs are explained.

Disassembly operation costs. These include the costs of all the operations designed to perform the disassembly process. This depends on the total time of the disassembly process and the cost per unit of time, as follows:

$$C_{OP} \;=\; t_T \, c_t \tag{4}$$

where:

- t_T is the total time spent by the robot to perform the disassembly process.
- c_t is the cost per unit of time.

The total time, t_T, to perform the disassembly process is determined as follows:

$$\begin{aligned} t_T \;&=\; \sum_{p=1}^{n} t_p \;+\; \sum_{p=1}^{n}\sum_{q=1}^{n} t_{p,q} \;= \\ &=\; \sum_{p=1}^{n} t_p \;+\; \sum_{p=1}^{n}\sum_{q=1}^{n} t_B(x_p, M, x_q)\,\gamma_{p,q} \;+\sum_{p=1}^{n}\sum_{q=1}^{n} t_C(x_p, x_q)\,(1-\gamma_{p,q}) \end{aligned} \tag{5}$$

where:

- t_p is the basic time to perform the disassembly operation x_p.
- $t_{p,q}$ is the total moving time between the disassembly points of adjacent operations x_p and x_q.
- $t_B(x_p, M, x_q)$ is the moving time between the disassembly points of adjacent operations x_p and x_q if the robot is required to change the tool in the tool magazine M.
- $\gamma_{p,q}$ is an indicator taking the value of 1 if operation x_q requires the robot to change the tool in tool magazine M once the previous operation x_p has been completed, and 0 otherwise.

- $t_C(x_p, x_q)$ is the moving time between the disassembly points of adjacent operations x_p and x_q if the robot is not required to change the tool in tool magazine M.

The moving time $t_B(x_p, M, x_q)$, if the robot is required to change the tool in tool magazine M, is defined as follows:

$$\begin{aligned} t_B(x_p, M, x_q) \;=\; & t_{B1}(x_p, M) \;+\; t_{B2}(M) \;+\; t_{B3}(M, x_q) + \\ & + t_{B4}(x_p, M) \;+\; t_{B5}(M, x_q) \end{aligned} \tag{6}$$

where:

- $t_{B1}(x_p, M)$ is the moving time between the disassembly point of the operation x_p and the tool magazine M. It is calculated by dividing the length between the disassembly point of x_p and M (obtained from the length matrix L), by line velocity v_e of the industrial robot's end-effect, as shown in Eq. 7.

  $$t_{B1}(x_p, M) \;=\; \frac{L(x_p, M)}{v_e} \tag{7}$$

  Length matrix $L = (l_{i,j})$ is a symmetric *(n+1)* x *(n+1)* matrix such that:

  - For *i=1,...,n* and *j=1,...,n*, $l_{i,j}$ is equal to the distance between the disassembly points i and *j*.
  - For *i=1,...,n*, $l_{i,(n+1)}$ is equal to the distance between the disassembly point i and tool magazine M.
  - For *j=1,...,n*, $l_{(n+1),j}$ is equal to the distance between tool magazine M and disassembly point *j*.
  - In particular, $l_{i,i}$=0 for all *i=1,...,n+1*

- $t_{B2}(M)$ is the tool change time the robot uses in tool magazine M and depends on the tool type. It is obtained from tool change matrix $MC =$

$(mc_{i,j})$, defined as a symmetric *n x n* matrix such that $mc_{i,j} = t_{B2}$ if the disassembly operation between item i and item j requires changing the tool in tool magazine M, and $mc_{i,j} = 0$ otherwise.

- $t_{B3}(M, x_q)$ is the moving time between tool magazine M and disassembly point of the operation x_q. It is calculated by dividing the length between M and the disassembly point of x_q (obtained from the length matrix L) by linear velocity v_e of the industrial robot's end-effect, as shown in Eq. 8.

$$t_{B3}(M, x_q) \ = \ \frac{L(M, x_q)}{v_e} \tag{8}$$

- $t_{B4}(x_p, M)$ is the penalty time for process direction changes along the path between disassembly point of x_p and tool magazine M. It is obtained from penalty time matrix $P = (p_{i,j})$, defined as a symmetric *(n+1)* x *(n+1)* matrix such that:

  - For *i=1,...,n* and *j=1,...,n*, $p_{i,j}$ is equal to $P_{t,1}$ if the direction between disassembly points i and j is changed by 90^o, $P_{t,2}$ if the direction is changed by 180^o, or zero if the direction is not changed.
  - For *i=1,...,n*, $p_{i,(n+1)}$ is equal to $P_{t,1}$ if the direction between disassembly point i and tool magazine M is changed by 90^o, $P_{t,2}$ if the direction is changed by 180^o, or zero if the direction is not changed.
  - For *i=1,...,n*, $p_{(n+1),j}$ is equal to $P_{t,1}$ if the direction between tool magazine M and disassembly point j is changed by 90^o, $P_{t,2}$ if the direction is changed by 180^o, or zero if the direction is not changed.

- $t_{B5}(M, x_q)$ is the penalty time for process direction changes along the path between M and the disassembly point of x_q. It is obtained from penalty time matrix P and formulated as $t_{B4}(x_p, M)$.

The calculation of the moving time if the robot is not required to change the tool in tool magazine M, is defined as follows:

$$t_C(x_p, x_q) = t_{C1}(x_p, x_q) + t_{C2}(x_p, x_q) \tag{9}$$

where:

- $t_{C1}(x_p, x_q)$ is the moving time between disassembly points of operations x_p and x_q. It is calculated by dividing the length between disassembly points of x_p and x_q (obtained from the L matrix) by line velocity v_e of the industrial robot's end-effect as shown in Eq. 10.

$$t_{C1}(x_p, x_q) = \frac{L(x_p, x_q)}{v_e} \tag{10}$$

- $t_{C2}(x_p, x_q)$ is the penalty time for process direction changes along the path between disassembly point of x_p and x_q operations. It is obtained from penalty time matrix P and is formulated as follows:
  - 0 if the direction is not changed.
  - $P_{t,1}$ if the direction is changed by 90^o.
  - $P_{t,2}$ if the direction is changed by 180^o.

  where $P_{t,1}$ and $P_{t,2}$ are expressed in units of time.

Disassembly recovery costs. These are the costs incurred in the operations to recover the components to be reused or remanufactured.

$$C_{RC} = \sum_{i=1}^{n} \sum_{m=1}^{2} rc_{i,m}\, \alpha_i \tag{11}$$

where:

- $rc_{i,m}$ is the recovery cost of the component i to be reused or remanufactured.

Disassembly overhead costs. These are the overhead costs including labour, rent, utilities, etc, and assigned to the disassembly process.

$$C_{OH} = \sum_{i=1}^{n} \sum_{m=1}^{4} oh_{i,m}\, \alpha_i \tag{12}$$

where:

- $oh_{i,m}$ is the overhead cost assigned to component i to be reused, remanufactured, recycled or disposed of.

Disassembly depreciation costs. These are the yearly depreciation costs of the machinery, considering a linear depreciation model.

$$C_{DP} = \sum_{i=1}^{n} \sum_{m=1}^{4} dp_{i,m}\, \alpha_i \tag{13}$$

where:

- $dp_{i,m}$ is the depreciation cost assigned to component i to be reused, remanufactured, recycled or disposed of.

Therefore, the objective function $f(X)$ assessing the economic profit in the disassembly process can be expressed as follows:

$$\begin{gathered} f(X) = \sum_{i=1}^{n} \sum_{m=1}^{2} RP_i r_{i,m} \alpha_i + \sum_{i=1}^{n} RR_i r_{i,3} \alpha_i - \sum_{i=1}^{n} CD_i r_{i,4} \alpha_i - \sum_{i=1}^{n} CD_i r_{i,4} (1-\alpha_i) - \\ - \sum_{p=1}^{n} t_p\, c_T\, \alpha_p - \sum_{p=1}^{n} \sum_{q=1}^{n} t_{p,q}\, c_T\, \delta_{p,q} - \\ - \sum_{i=1}^{n} \sum_{m=1}^{2} rc_{i,m}\, r_{i,m}\, \alpha_i - \sum_{i=1}^{n} \sum_{m=1}^{4} oh_{i,m}\, r_{i,m}\, \alpha_i - \sum_{i=1}^{n} \sum_{m=1}^{4} dp_{i,m}\, r_{i,m}\, \alpha_i \end{gathered} \tag{14}$$

where:

- α_p is an indicator taking value 1 if operation x_q must be completed in the disassembly process, and 0 otherwise.
- $\delta_{p,q}$ is an indicator taking value 1 if operation x_q is carried out after operation x_p, and 0 otherwise.

3.2.3. Constraints

Some constraints have to be considered in the disassembly process as shown in the following equations:

$$\sum_{m=1}^{4} r_{i,m} \; = \; 1 \qquad \forall i \tag{15}$$

$$r_{i,1} \; + \; r_{i,2} \; + \; r_{i,3} \; \leq \; \alpha_i \tag{16}$$

where Eq. 15 guarantees that each component i has only one recovery mode, and Eq. 16 ensures that all components to be reused, remanufactured or recycled must be disassembled.

4. Algorithms

The model formulated is an NP-complete and intractable problem to be exactly computed even for medium-size problem instances. As in related studies on disassembly (Meng et al., 2017; Rickli and Camelio, 2013; Zhang et al., 2017), we propose managing the problem with approximate algorithms, specifically, heuristics. We use a constructive greedy heuristic, a local search-based metaheuristic and a global search-based metaheuristic (Gendreau and Potvin, 2010; Michalewicz and Fogel, 2013).

4.1. Problem formulation

Let $I = \{1, 2, \ldots, n\}$ be the set of pieces to be disassembled. For every $j \in I$ let us denote by

$$P(j) = \{i \in I : i \text{ must be disassembled before } j\}$$

the set of pieces which need to be disassembled before j. We assume that for every pair of items $i, j \in I$, if $i \in P(j)$ then $j \notin P(i)$. Note that it can be $i \notin P(j)$ and $j \notin P(i)$.

Let us consider the matrix $n \times n$ of precedences $C = (c_{i,j})$ defined by

$$c_{i,j} = 1 \quad \text{if } i \in P(j)$$

$$c_{i,j} = 0 \quad \text{if } i \notin P(j)$$

Namely, the 1's in the j-th column correspond to the pieces that must be disassembled before piece j. Observe that

$$\forall i, j, k \in I, \quad c_{i,j} = 1 \, \text{and} \, c_{j,k} = 1 \implies c_{i,k} = 1.$$

Given a permutation π of the elements in I, we will write $j \prec_\pi i$ to express that j precedes i in π, where $i, j \in I$. We will say that π is compatible with (the precedences expressed in) C if

$$\forall i, j \in I, \quad j \prec_\pi i \, \rightarrow \, c_{i,j} = 0$$

or, equivalently,

$$\forall i, j \in I, \quad j \prec_\pi i \, \rightarrow \, i \notin P(j).$$

We will write π_i to denote the i-th element in the permutation π.

Associated with every $i \in I$, we consider the variables $r_{i,m} \in \{0, 1\}$, $m \in \{1, 2, 3, 4\}$, with $\sum r_{i,m} = 1$. Moreover, given a permutation π compatible with C, we will call integer $t(\pi)$ the threshold of π, $0 \leq t(\pi) \leq n$, meaning that the items ordered in π from the $t(\pi)$-th position onwards are not disassembled. In particular, if $t(\pi) = 0$, no piece is disassembled, and if $t(\pi) = n$ all the pieces of the product are disassembled. Observe that $r_{i,4} = 1$ for any item i ordered after the $t(\pi)$-th position.

From the previous description, a solution to our problem is a triplet $\langle \pi, v, t \rangle$, where:

- π is a permutation compatible with a precedence matrix C that indicates the order in which the pieces must be disassembled.

- v is a vector of length n, containing for each $i \in I$ the value $m \in \{1, \ldots, 4\}$ such that $r_{i,m} = 1$, that is, the destination of piece π_i.
- t is the threshold $t(\pi)$ of π, i.e. the position in which the disassembling process is stopped.

Accordingly, the cardinality of the search space is $n! \times 4^n \times n$. In fact, this is an upper bound because not all the permutations are compatible with C. However, for our particular problem and objective function $f(X)$, we can reduce this search space by pre/post-computing some values. We can observe from the expression of $f(X)$ in Eq. 14 that the most profitable destination for each piece is independent of the destination of the other pieces and the order in which they are disassembled. Therefore, vector v can be precomputed before seeking the most profitable disassembly order. Note also that the model allows the destination of some pieces to be given as input.

Once we know v, we reduce the optimisation problem to the search for disassembly order π and threshold t. In fact, these two values do not need to be obtained simultaneously, because the best t value can be obtained for a given ordering by inspecting all the $(n+1)$ possible thresholds and selecting the most profitable one.

Therefore, our optimisation problem can be reduced to the search for the most profitable order compatible with precedence matrix C. Once the order is obtained, we associate with it the $f(X)$ value corresponding to its best threshold. We propose some algorithms to compute this order in the following subsections.

4.2. Constructive Greedy Algorithm

This involves incrementally constructing a solution (permutation) by choosing, position by position, the item which most adds to the objective function $f(X)$ being compatible with the precedences established by the items already ordered. Note that the value *added* by each piece depends on the piece previously disassembled, which sets the current location and tool.

Let π_r denote a partial solution (incomplete sequence), which only includes the first r pieces to be disassembled and let $\pi_r \circ i$ denote the partial solution of size $r + 1$ formed by adding piece i at the end of π_r. Then, we can use a greedy algorithm to determine the next piece to be disassembled by trying any remaining piece, which may already be disassembled, and choosing the one maximising the f-value. Formally:

$$i^* = \arg \max_{i \in I \setminus \pi_r \,\wedge\, s(\pi_r) \subseteq P(i)} f(\pi_r \circ i), \tag{17}$$

where $s(\pi_r)$ is the set containing the pieces included in π_r.

The pseudo-code of the constructive greedy algorithm is shown in Figure 2.

```
CGA ($f$, $I$)
input: $I$, the set of $n$ pieces to be disassembled
input: $f$, the objective function to be maximised
input: $P$, the precedence function
$\pi_r \leftarrow$ empty
$I' \leftarrow I$
while $I' \neq \emptyset$ do
    $i^* = \arg\max_{i \in I' \,\wedge\, s(\pi_r) \subseteq P(i)} f(\pi_r \circ i)$
    $\pi_r \leftarrow \pi_r \circ i^*$
    $I' \leftarrow I' \setminus \{i^*\}$
endwhile
return $\pi_r$
```

Figure 2: Constructive Greedy Algorithm.

4.3. Hill Climbing Algorithm

The HCA (Selman and Gomes, 2006) is a well-known local-search method that attempts to improve a given solution by analysing those in the close neighbourhood. A solution is said to be a neighbour of a given solution if it can be obtained from this by applying a certain *mutation* operator, that is, an operator that introduces a *small* change in the given solution. In the case of permutation-based optimisation problems, one of the most commonly used mutation operators is the *Interchange* one. However, although this operator is *closed* in the space of permutations, it is not valid in our case, because we also need the resulting permutation to be compatible with precedence matrix C. Therefore, we define the following *restricted* interchange operator:

Restricted Interchange Mutation $\mathcal{M}^C$: Let π be a permutation compatible with a precedence matrix $C = (c_{ij})$, i.e. $c_{ji} = 0$ for all $i, j \in I$ such that $i \prec_\pi j$. Take $i, j \in I$, $i \prec_\pi j$ verifying

- $c_{i,j} = 0$
- $c_{i,k} = c_{k,j} = 0$ for all $k \in I$ such that $i \prec_\pi k \prec_\pi j$

If we consider π' as the permutation obtained from π by interchanging items i and j, it is simple to check that π' is compatible with C. In fact, note that for $k \in I$ such that $i \prec_\pi k \prec_\pi j$, we have that $j \prec_{\pi'} k \prec_{\pi'} i$ and the compatibility follows from the required conditions $c_{i,j} = 0$ and $c_{i,k} = c_{k,j} = 0$.

By using this operator, we can define the neighbourhood used in our approach:

$$\mathcal{N}(\pi) = \{\pi' : \pi' \text{ can be obtained from } \pi \text{ by applying mutation } \mathcal{M}^C\}$$

The method is then applied. We start with an arbitrary solution, generate the neighbours and evaluate them according to $f(X)$. If the best neighbour is better than the current solution, we *move* to it and iterate again; otherwise, the method stops returning the current solution. In Figure 3, we show the pseudo-code of the hill climbing algorithm:

One of the advantages of using HCA is that the solution obtained is guaranteed to be a local optimum for the given operator. The disadvantage is that the method can be quickly trapped in a local optimum that can be far from the (global) optimum. To mitigate this disadvantage, we use HCA with re-starts, a standard way to attempt to escape from local optima (Lourenço et al., 2002). In particular, we run the HCA 50 times starting from (different) randomly generated solutions.

```
HCA ($\pi$, $f$)
input: $\pi$, the starting point or initial candidate solution.
input: $f$, the objective function to be maximised
$f_\pi \leftarrow f(\pi)$                    // evaluate $\pi$
$improving \leftarrow true$
Repeat
    Compute the neighborhood $\mathcal{N}(\pi)$
    $\pi^* \leftarrow \arg\max_{\pi' \in \mathcal{N}(\pi)} f(\pi')$
    if $f(\pi^*) < f_\pi$ then
        $(\pi, f_\pi) \leftarrow (\pi^*, f(\pi^*))$
    endif
until $\pi$ does not change
return $(\pi, f_\pi)$
```

Figure 3: Hill Climbing Algorithm (maximising).

4.4. Genetic Algorithm

As previously mentioned, one of the disadvantages of HCA is that it can be quickly trapped in a local optimum. An alternative to overcome this problem is to use global search algorithms, which usually consider a population of solutions to explore the search space. This is the case of Genetic Algorithms (GA) (Michalewicz, 1992; Michalewicz and Fogel, 2013), a bio-inspired method whose success has been demonstrated when applied to a wide range of real-world problems.

In a GA, a population of solutions is initially generated and evaluated. Then, a selection process is carried out by giving more probability of selection to solutions having a better score. The selected solutions are then recombined by using two genetic operators: crossover and mutation, giving rise to a population of offsprings. The new individuals are then evaluated and the new population is obtained by combining the previous and the generated populations in some way. In Figure 4, we show the scheme of a canonical GA.

Now, let us describe how we have customised the different operations to our

```
GA(operators, parameters)
input: Genetic operators (selection, crossover, mutation)
input: Parameters (pop. size, crossover and mutation probabilities, $p_c$ and $p_m$)
Create and evaluate initial population $P_0$
$t \leftarrow 1$
While not stopping criterion do
    $P_{sel} \leftarrow$ Select parents from $P_{t-1}$
    $P_{cross} \leftarrow$ Get offsprings by applying crossover with prob. $p_c$ over $P_{sel}$
    $P_{mut} \leftarrow$ Apply mutation operator with prob. $p_m$ over $P_{cross}$
    $P_t \leftarrow$ Get population from $P_{t-1} \cup P_{mut}$
    $t \leftarrow t + 1$
end
return best individual found during the search
```

Figure 4: Genetic Algorithm pseudocode

problem:

- *Population.* As is customary, we set the population size proportional to the problem dimension n (i.e. number of pieces to be disassembled). In particular, we consider a population of $10n$ individuals, a common value in the literature.

- *Population initialisation.* We initialise the population randomly, but take care to only include valid solutions, that is, permutations compatible with C.

- *Selection.* In order to maintain diversity, a tournament selection (Miller et al., 1995) of size 2 is used. Specifically, at each iteration, $10n$ pairs of individuals are randomly chosen and the individual of each pair with best fitness, $f(X)$ value, is selected.

- *Crossover and mutation operators.* As mutation operator, we use operator $\mathcal{M}^C$ defined in the previous section. As crossover, we use a classic operator to deal with permutations (Larranaga et al., 1999), which in our case also has the advantage of being closed given C, that is, if the parent solutions are compatible with C, then the two children obtained are also compatible with C. Specifically, let π and σ be two permutations (parents) compatible with a precedence matrix C, and take two cut positions p_1 and p_2, $0 \leq p_1 < p_2 \leq n$. Then two new permutations (children) π' and σ' compatible with C are obtained as follows:

- π' has the first p_1 and the last $n - p_2$ items ordered as in π, while the remaining items are ordered, according to their relative order in σ, in positions $p_1 + 1, \ldots, p_2$.
- σ' has the first p_1 and the last $n - p_2$ items ordered as in σ, while the remaining items are ordered, according to their relative order in π, in positions $p_1 + 1, \ldots, p_2$.

For example, take $n = 6$, $\pi = (1, 2, 3, 4, 5, 6)$, $\sigma = (3, 2, 5, 4, 6, 1)$, $p_1 = 2$ and $p_2 = 5$. Then $\pi' = (1, 2, 3, 5, 4, 6)$ and $\sigma' = (3, 2, 4, 5, 6, 1)$.

The recombination phase is applied as follows. Solutions are set in pairs as they are obtained in the selection phase. Then, crossover is applied to each pair (i.e. crossover probability is 1). For each child, we decide if it must be muted (by applying $\mathcal{M}^C$) or not according to a given mutation probability (0.15 in our case).

- *Next Population Construction.* To obtain the population for the next generation a *truncation* operator is used. Specifically, the offspring population and the current population are put together in a common pool, and the best $10n$ individuals are selected.
- *Stopping criterion.* The algorithm stops executing when we consider that it has converged. In this study, we stop after a given maximum number of generations (100). In particular, we have observed that in this problem, when sufficient diversity is allowed in the population (e.g. $10n$ individuals), the method always obtains the best solution of each run in a generation ranged from the 10-th to the 20-th one.

5. Experimental application

5.1. Gear pump

The gear pump is extensively used in industry for the transfer and displacement of fluids. Essentially, it consists of two gears managed by two axes and enclosed in a tight housing. It transforms kinetic energy in the form of torque, generated by a motor, into hydraulic energy through the flow of oil generated by the pump. This flow of pressurised oil is used to generate the movement of the actuator installed in the machine or application. In this paper, a gear pump of 10 l/min, as shown in Figure 5 (Grabcad Community, 2018) is used as the experimental application. Figure 6 shows an exploded view with all its components.

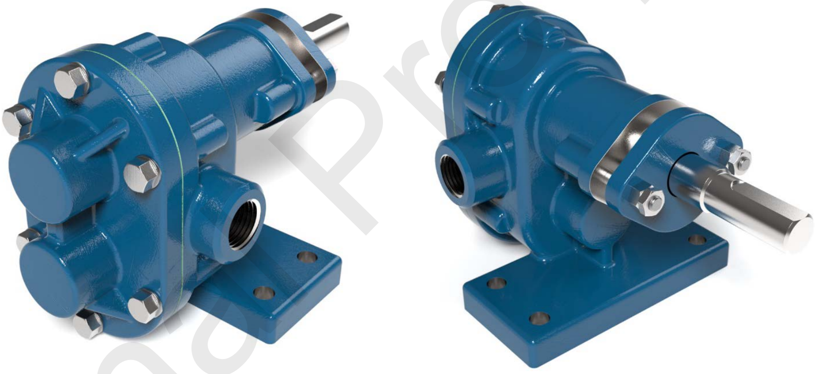


Figure 5: Gear pump. External view (Source: Grabcad Community (2018))

From the perspective of component recovery, and based on the information from manufacturers, some items may have new applications depending on the state and the quality of each component. In this way, some components could be reused or remanufactured (7, 9, 10, 11, 12, 13 and 18), and the other components could be recycled or ultimately disposed of.

Table 1 shows the properties of the components and the precedence relationship in the disassembly process between the components. The information was

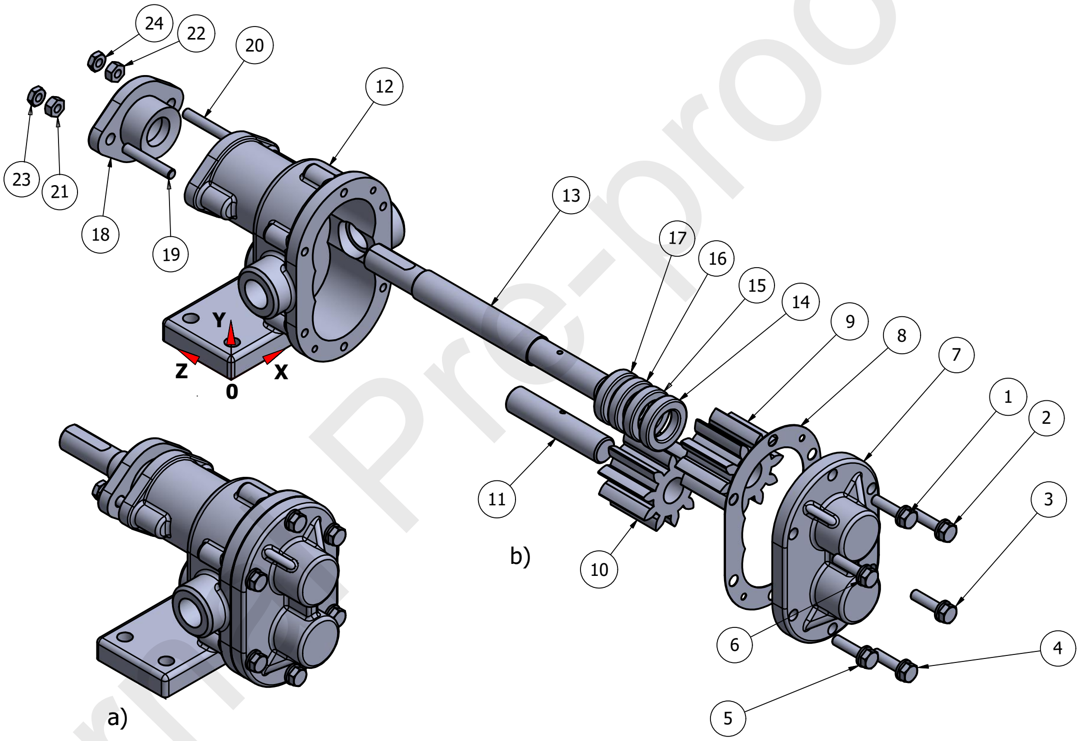


Figure 6: Gear pump: a) Assembled view; b) Exploded view (Source: Grabcad Community (2018))

obtained from manufacturers, the 3D model (Grabcad Community, 2018) and the work of Liu et al. (2018).

Table 1: Gear pump. Properties of all components.

No.	Component	Material	Volume (mm^3)	Weight (g)	Predecessors
1	Bolt A	Steel	1,243.07	9.76	-
2	Bolt B	Steel	1,243.07	9.76	-
3	Bolt C	Steel	1,243.07	9.76	-
4	Bolt D	Steel	1,243.07	9.76	-
5	Bolt E	Steel	1,243.07	9.76	-
6	Bolt F	Steel	1,243.07	9.76	-
7	Cover	Steel	95,973.49	753.39	1-6
8	Gasket	Rubber	5,496.27	5.22	1-7
9	Gear A	Steel	21,301.72	167.22	1-8
10	Gear B	Steel	21,301.72	167.22	1-8
11	Shaft A	Steel	6,430.70	50.48	1-8,10
12	Base	Steel	273,754.96	2,148.98	1-11,13-24
13	Shaft B	Steel	22,560.02	177.10	14-24
14	Gland A	PTFE	3,243.59	7.14	15-24
15	Gland B	PTFE	3,243.59	7.14	16-24
16	Gland C	PTFE	3,243.59	7.14	17-24
17	Gland D	PTFE	3,243.59	7.14	18-24
18	Gland E	Steel	14,456.27	113.48	19-24
19	Bolt stud A	Steel	998.08	7.83	21,23
20	Bolt stud B	Steel	998.08	7.83	22,24
21	Nut A	Steel	289.52	2.27	23
22	Nut B	Steel	289.52	2.27	24
23	Nut C	Steel	289.52	2.27	-
24	Nut D	Steel	289.52	2.27	-

5.2. Robot and robotic tools

The industrial automation system proposed in the simulations consists of two units: the robot and the robotic tools. Figure 7 shows the layout approach of the robotic cell used for the simulations, with the location of the robot, the gear pump, and the tools magazine.

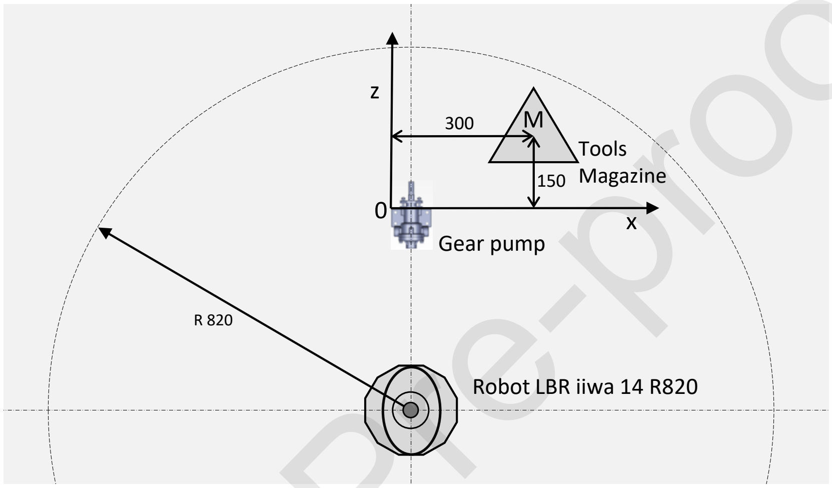


Figure 7: Robotic cell layout

The KUKA LBR iiwa 14 R820 was selected as the robot to carry out the simulations. It is classified as a lightweight robot with a jointed-arm with 7 axes. Table 2 shows the basic data for this robot (KUKA, 2019).

Table 2: Basic data, LBR iiwa 14 R820

Number of axes	7
Number of controlled axes	7
Volume of working envelope	1.8 m^3
Pose repeatability (ISO 9283)	±0.15 mm
Weight approx.	29.9 kg
Rated payload	14 kg
Maximum reach	820 mm

Concerning the operating parameters of the robot, and based on the information provided by the manufacturer, certain assumptions are considered, as follows:

- The linear velocity of the robot's end-effector is assumed to be 12 mm/s.
- A safe distance along the contour of the product is considered in order to allow the robot to avoid impractical paths between the disassembly points and the tool magazine M.
- The time the robot takes to change the tool in the tool magazine M, t_{B2}, is assumed to be 10 seconds.
- The penalty time for process direction changes t_{C2} is assumed to be 0 if the direction is not changed, 1 second if the direction is changed by 90°, and 2 seconds if the direction is changed by 180°, according to the axis data provided in the robot's technical specifications (KUKA, 2019).

Robotic tools such as cutters, drillers and grippers are the execution units, while the robots move robotic tools to the requested positions. Changing operations may require changes of robotic tools. There are two main disassembly operations in the case of dismantling gear pumps: unfastening and pulling/pushing using a gripper. Due to the different types of components and sizes, three types of spanners and two types of grippers are used. Table 3 shows the disassembly tools required to complete each of the operations, and the basic time (t_p) required to perform the disassembly operation. It also shows the coordinates of the disassembly points referring to the coordinates of the origin as shown in Figure 7.

Additionally, the layout includes a tool magazine M. This is a device containing the tools needed to complete the disassembly process. The robot moves up to the position of the tool magazine M if the subsequent planned disassembly operation requires changing the tool. In our case, the tool magazine M is assumed to be located at position x=300, y=0, z=150, according to Figure 7.

Table 3: Gear pump. Disassembly points and robotic tools

No.	**Component**	**Disassembly point**			**Disassembly**	t_p
		X	**Y**	**Z**	tool	(s)
1	Bolt A	59.1	114	-48.4	Spanner-I	4
2	Bolt B	90.3	89	-48.4	Spanner-I	4
3	Bolt C	90.3	33	-48.4	Spanner-I	4
4	Bolt D	59.1	8	-48.4	Spanner-I	4
5	Bolt E	27.9	33	-48.4	Spanner-I	4
6	Bolt F	27.9	89	-48.4	Spanner-I	4
7	Cover	59.1	82	-64.6	Gripper-II	5
8	Gasket	59.1	114	-31.4	Gripper-I	4
9	Gear A	59.1	82	-30.9	Gripper-I	6
10	Gear B	59.1	40	-30.9	Gripper-I	6
11	Shaft A	59.1	40	-48.9	Gripper-I	4
12	Base	59.1	114	7.1	Gripper-II	4
13	Shaft B	59.1	82	136.1	Gripper-I	8
14	Gland A	59.1	94.8	34.1	Gripper-I	3
15	Gland B	59.1	94.8	41.1	Gripper-I	3
16	Gland C	59.1	94.8	48.1	Gripper-I	3
17	Gland D	59.1	94.8	55.1	Gripper-I	3
18	Gland E	59.1	82	79.1	Gripper-I	3
19	Bolt stud A	35.1	82	89.1	Spanner-II	3
20	Bolt stud B	83.1	82	89.1	Spanner-II	3
21	Nut A	35.1	82	84.1	Spanner-III	4
22	Nut B	83.1	82	84.1	Spanner-III	4
23	Nut C	35.1	82	87.1	Spanner-III	4
24	Nut D	83.1	82	87.1	Spanner-III	4

The process flow in the sequence of disassembly adopted in this case is depicted in Figure 8.

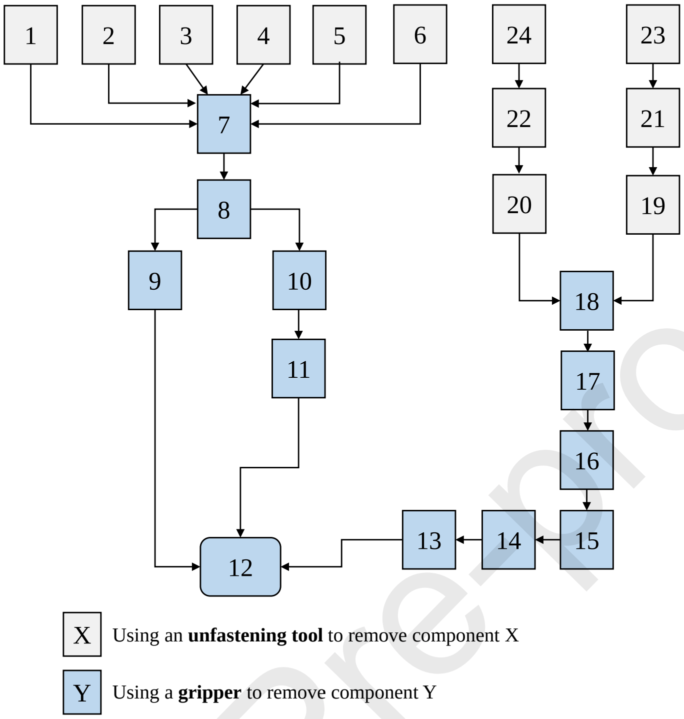

1
2
3
4
5
6
24
23
7
22
21
8
20
19
9
10
18
11
17
16
13
14
15
12
X
Using an **unfastening tool** to remove component X
Y
Using a **gripper** to remove component Y

Figure 8: Gear pump. Disassembly process flow.

Table 4 shows a matrix summarising the values of the $t_{p,q}$ parameter (Eq. 5), taking into account all the feasible paths between disassembly points. It is important to highlight that the moving path between the disassembly points was obtained using the Euclidean distance (Alshibli et al., 2016), considering a safe distance along the contour of the product and allowing the robot to avoid impractical paths between disassembly points due to the obstacles caused by the contour of the product being disassembled.

Table 4: Matrix of moving time between adjacent disassembly points ($t_{p,q}$)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0.00	6.35	11.02	10.50	11.02	6.35	51.20	Inf	Inf	Inf	Inf	51.50	57.08	52.52	52.74	52.98	53.23	54.40	56.49	53.21	56.28	52.98	56.40
2	6.35	0.00	6.33	11.02	11.53	6.87	49.11	Inf	Inf	Inf	Inf	49.41	54.99	50.43	50.65	50.88	51.13	52.30	54.39	51.12	54.19	50.88	54.31
3	11.02	6.33	0.00	6.35	6.87	11.53	50.92	Inf	Inf	Inf	Inf	51.22	56.79	52.23	52.46	52.69	52.94	54.11	56.20	52.92	56.00	52.69	56.12
4	10.50	11.02	6.35	0.00	6.35	11.02	54.20	Inf	Inf	Inf	Inf	54.50	60.08	55.52	55.74	55.98	56.23	57.39	59.49	56.21	59.28	55.98	59.40
5	11.02	11.53	6.87	6.35	0.00	6.33	55.58	Inf	Inf	Inf	Inf	55.88	61.46	56.90	57.12	57.35	57.60	58.77	60.86	57.59	60.66	57.35	60.78
6	6.35	6.87	11.53	11.02	6.33	0.00	54.14	Inf	Inf	Inf	Inf	54.44	60.01	55.45	55.68	55.91	56.16	57.33	59.42	56.14	59.22	55.91	59.34
7	Inf	Inf	Inf	Inf	Inf	Inf	0.00	51.35	Inf	Inf	Inf	10.31	57.60	53.04	53.26	53.50	53.74	54.91	57.00	53.73	56.80	53.50	56.92
8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0.00	4.38	7.88	Inf	51.64	18.29	8.73	9.31	9.89	10.48	13.54	16.38	53.35	56.43	53.12	56.55
9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0.00	5.17	Inf	52.21	15.58	8.15	8.73	9.32	9.90	10.83	13.67	53.91	56.99	53.68	57.11
10	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	5.17	0.00	3.17	53.41	19.08	11.65	12.23	12.82	13.40	14.33	17.17	55.11	58.19	54.88	58.31
11	52.45	50.36	52.16	55.45	56.83	55.38	52.97	9.29	6.67	3.17	0.00	53.27	20.58	13.15	13.73	14.32	14.90	15.83	18.67	54.97	58.05	54.74	58.17
12	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0.00	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
13	57.08	54.99	56.79	60.08	61.46	60.01	57.60	18.29	15.58	19.08	20.58	57.90	0.00	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
14	52.52	50.43	52.23	55.52	56.90	55.45	53.04	8.73	8.15	11.65	13.15	53.34	11.23	0.00	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
15	52.74	50.65	52.46	55.74	57.12	55.68	53.26	9.31	8.73	12.23	13.73	53.56	Inf	2.25	0.00	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
16	52.98	50.88	52.69	55.98	57.35	55.91	53.50	9.89	9.32	12.82	14.32	53.79	Inf	Inf	2.25	0.00	Inf	Inf	Inf	Inf	Inf	Inf	Inf
17	53.23	51.13	52.94	56.23	57.60	56.16	53.74	10.48	9.90	13.40	14.90	54.04	Inf	Inf	Inf	2.25	0.00	Inf	Inf	Inf	Inf	Inf	Inf
18	54.40	52.30	54.11	57.39	58.77	57.33	54.91	13.54	10.83	14.33	15.83	55.21	Inf	Inf	Inf	Inf	4.73	0.00	Inf	Inf	Inf	Inf	Inf
19	56.49	54.39	56.20	59.49	60.86	59.42	57.00	56.63	57.19	58.39	58.25	57.30	Inf	Inf	Inf	Inf	Inf	60.20	0.00	5.67	Inf	58.78	Inf
20	53.21	51.12	52.92	56.21	57.59	56.14	53.73	53.35	53.91	55.11	54.97	54.03	Inf	Inf	Inf	Inf	Inf	56.92	5.67	0.00	58.81	Inf	58.93
21	56.28	54.19	56.00	59.28	60.66	59.22	56.80	56.43	56.99	58.19	58.05	57.10	Inf	Inf	Inf	Inf	Inf	Inf	62.08	58.81	0.00	5.67	Inf
22	52.98	50.88	52.69	55.98	57.35	55.91	53.50	53.12	53.68	54.88	54.74	53.79	Inf	Inf	Inf	Inf	Inf	Inf	58.78	55.50	5.67	0.00	5.92
23	56.40	54.31	56.12	59.40	60.78	59.34	56.92	56.55	57.11	58.31	58.17	57.22	Inf	Inf	Inf	Inf	Inf	Inf	Inf	58.93	1.92	5.92	0.00
24	53.12	51.02	52.83	56.12	57.49	56.05	53.63	53.26	53.82	55.02	54.88	53.93	Inf	Inf	Inf	Inf	Inf	Inf	58.92	Inf	5.92	1.92	5.67

Inf: Infeasible path between disassembly points

5.3. Economic parameters

Concerning the economic parameters related to the disassembly process, Table 5 presents the revenues and costs associated with each component to be disassembled. In their assessment, a number of considerations and assumptions were taken into account, as follows:

- The cost per unit of time c_t is assumed as 0.05 €/s (180 €/h), according to the information from manufacturers in Spain in July 2018.
- The revenue obtained from each component to be reused or remanufactured (RP_i) was obtained from the information supplied by manufacturers.
- The revenue obtained from each component to be recycled (RR_i) was obtained from recyclers according to the material and weight of each component.
- The disposal cost was evaluated by the authors depending on the material and weight of the component to be disposed of, considering the most appropriate, environmental mode of disposal.
- The recovery costs of the components to be reused or remanufactured ($rc_{i,m}$) were calculated by the authors with technical information provided by manufacturers and the industry.
- The overhead costs of the company were allocated to each component according to the disassembly operation to be performed and the different recovery alternatives (reuse, remanufacturing, recycling or disposal), and also considering the following:
  - Total yearly overhead costs of the company of 240,000 €
  - The company operates 220 days per year, 8 hours per day.
  - The forecast production of disassembled gear pumps is 52,800 units per year.

  - The distribution of the overhead costs to each component was evaluated taking into account the company's resources applied to the disassembly process, and according to the final use of each component. In this way, a weight over 10 was assigned to each recovery mode as follows: 2 for reuse, 5 for remanufacturing, 2 for recycling, and 1 for disposal.

- The depreciation costs of the machinery were assessed taking into account an estimated investment of 1 M€. The model considers linear depreciation over a period of 10 years. The depreciation costs were assigned to each component considering also the disassembly time of each component and the moving time between adjacent disassembly points.

Table 5: Revenues and costs associated with all components (€).

Item	RP_i	RR_i	CD_i	$rc_{i,1}$	$rc_{i,2}$	$oh_{i,1}$	$oh_{i,2}$	$oh_{i,3}$	$oh_{i,4}$	$dp_{i,1}$	$dp_{i,2}$	$dp_{i,3}$	$dp_{i,4}$
1	0.432	0.005	0.000	0.100	0.300	0.002	0.006	0.002	0.001	0.115	0.173	0.138	0.092
2	0.432	0.005	0.000	0.100	0.300	0.002	0.006	0.002	0.001	0.115	0.173	0.138	0.092
3	0.432	0.005	0.000	0.100	0.300	0.002	0.006	0.002	0.001	0.115	0.173	0.138	0.092
4	0.432	0.005	0.000	0.100	0.300	0.002	0.006	0.002	0.001	0.115	0.173	0.138	0.092
5	0.432	0.005	0.000	0.100	0.300	0.002	0.006	0.002	0.001	0.115	0.173	0.138	0.092
6	0.432	0.005	0.000	0.100	0.300	0.002	0.006	0.002	0.001	0.115	0.173	0.138	0.092
7	7.900	0.377	0.000	1.200	2.500	0.185	0.463	0.185	0.093	0.144	0.216	0.173	0.115
8	0.000	0.003	0.200	0.000	0.000	0.001	0.003	0.001	0.001	0.115	0.173	0.138	0.092
9	10.780	0.084	0.000	0.600	3.000	0.041	0.103	0.041	0.021	0.173	0.259	0.207	0.138
10	10.780	0.084	0.000	0.600	3.000	0.041	0.103	0.041	0.021	0.173	0.259	0.207	0.138
11	2.320	0.025	0.000	0.300	0.900	0.012	0.031	0.012	0.006	0.115	0.173	0.138	0.092
12	11.800	1.074	0.000	1.500	3.500	0.529	1.322	0.529	0.264	0.115	0.173	0.138	0.092
13	5.780	0.089	0.000	0.700	2.000	0.044	0.109	0.044	0.022	0.230	0.346	0.276	0.184
14	0.000	0.004	0.150	0.000	0.000	0.002	0.004	0.002	0.001	0.086	0.130	0.104	0.069
15	0.000	0.004	0.150	0.000	0.000	0.002	0.004	0.002	0.001	0.086	0.130	0.104	0.069
16	0.000	0.004	0.150	0.000	0.000	0.002	0.004	0.002	0.001	0.086	0.130	0.104	0.069
17	0.000	0.004	0.150	0.000	0.000	0.002	0.004	0.002	0.001	0.086	0.130	0.104	0.069
18	3.600	0.057	0.000	0.200	0.700	0.028	0.070	0.028	0.014	0.086	0.130	0.104	0.069
19	0.350	0.004	0.000	0.100	0.300	0.002	0.005	0.002	0.001	0.086	0.130	0.104	0.069
20	0.350	0.004	0.000	0.100	0.300	0.002	0.005	0.002	0.001	0.086	0.130	0.104	0.069
21	0.200	0.001	0.000	0.100	0.300	0.001	0.001	0.001	0.000	0.115	0.173	0.138	0.092
22	0.200	0.001	0.000	0.100	0.300	0.001	0.001	0.001	0.000	0.115	0.173	0.138	0.092
23	0.200	0.001	0.000	0.100	0.300	0.001	0.001	0.001	0.000	0.115	0.173	0.138	0.092
24	0.200	0.001	0.000	0.100	0.300	0.001	0.001	0.001	0.000	0.115	0.173	0.138	0.092

6. Results and discussion

Simulations were performed using an iMac computer with an Intel Core i5 (3.2 GHz) processor and running operating system OS X 10.11.6. The algorithms were programmed in Java and run using the JVM 1.8. In the disassembly problem studied in this paper, the cardinality of the search space is bounded by $24! \approx 6.2 \times 10^{23}$. Each simulation was repeated 100 times for the non-deterministic algorithms. In the case of non-deterministic algorithms, different seeds were used to initiate the search.

The results are divided into three subsections. The first subsection shows the optimal solution obtained using the proposed algorithms for several iterations and populations. The second subsection presents solutions considering different recovery approaches. The third subsection discusses a sensitivity analysis performed using the model in order to evaluate how the uncertainty behaviour of selected variables could affect the economic performance of the disassembly process.

6.1. Optimal solution

Table 6 shows the optimal solution results for the three calculation algorithms considered, which propose a complete disassembly of the product, although the disassembly sequence is different for each one. The best solution is achieved by the GA, with a fitness value of 20.56 €, followed by CGA (20.05 €) and HCA (19.28 €). The three solutions recommend all the parts be disassembled and reused, with the exception of items 15, 16, 17, 18 and 20, which should be sent for disposal.

Figure 9 shows the evolution of the fitness value according to the disassembly level for the three studied simulations. The best performance is achieved by the GA, obtaining positive profits after the 16th disassembly operation (item n^{o} 11). The HCA solution obtains positive profits after the 17th operation (item n^{o} 9) and, in the case of CGA, positive profit is achieved after the 21th operation.

Table 6: Optimal solution results

CGA	DS level	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
	Item No.		2	3	4	1	6	5	7	24	22	23	21	20	19	18	17	16	15	14	13	8	9	10	11
	Recovery mode		1	1	1	1	1	1	1	1	1	1	1	1	1	1	4	4	4	4	1	4	1	1	1
	Fitness value (€)	-0.80	-2.62	-3.02	-3.49	-3.86	-4.31	-4.69	-1.15	-3.95	-4.26	-4.95	-5.26	-8.04	-8.48	-8.10	-8.50	-8.82	-9.15	-9.62	-6.00	-7.09	2.33	11.68	13.21
	Time (ms/run)	32																							
HCA	DS level	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
	Item No.		4	2	3	5	1	6	23	24	21	22	19	20	7	8	10	11	9	18	17	16	15	14	13
	Recovery mode		1	1	1	1	1	1	1	1	1	1	1	1	1	4	1	1	1	1	4	4	4	4	1
	Fitness value (€)	-0.80	-3.13	-3.41	-3.81	-4.38	-4.71	-5.16	-8.49	-8.63	-9.31	-9.64	-12.75	-12.86	-9.32	-12.21	-2.88	-1.19	8.35	10.79	10.24	9.77	9.30	8.82	12.44
	Time (ms/run)	7.8																							
GA	DS level	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
	Item No.		23	21	24	22	2	1	6	5	4	3	7	19	20	8	10	11	9	18	17	16	15	14	13
	Recovery mode		1	1	1	1	1	1	1	1	1	1	1	1	1	4	1	1	1	1	4	4	4	4	1
	Fitness value (€)	-0.80	-3.63	-3.94	-4.30	-4.61	-6.92	-7.34	-7.79	-7.97	-8.22	-8.36	-4.82	-7.93	-8.04	-10.93	-1.59	0.09	9.63	12.07	11.52	11.05	10.58	10.11	13.73
	Time (ms/run)	37																							

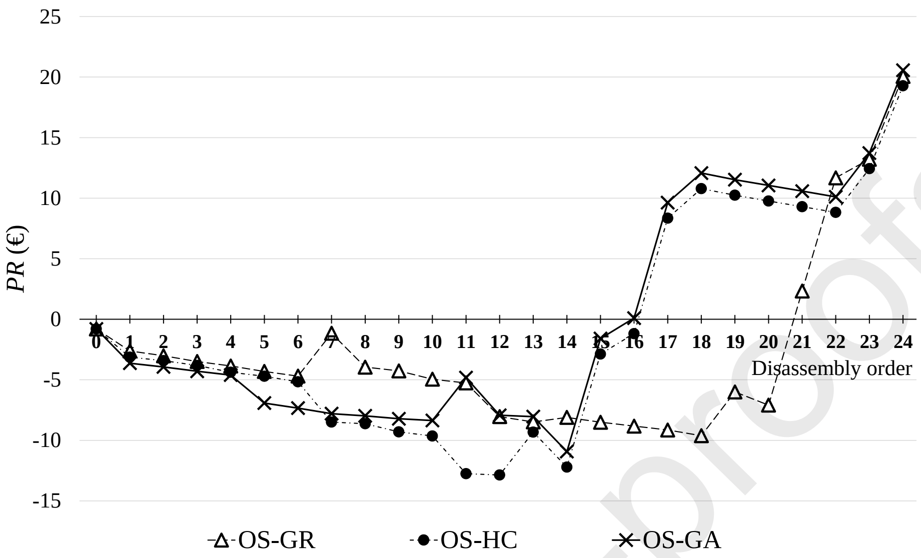


Figure 9: Optimal solution results. Economic profit *(PR)* in each *-th* position of the disassembly process for the three algorithms.

6.2. Results according to recovery approaches

The condition of the components to be disassembled is diverse and depends on the use and how they have been handled by the customers, and also because the product naturally deteriorates over its lifetime. In order to analyse how the condition of the components could make an impact on the performance of the robotic disassembly process, this subsection presents the results considering three alternative recovery approaches for the components: reuse (REU), remanufacturing (REM) and recycling (REC).

From the perspective of economic assessment, these alternatives entail the following considerations:

- REU is the most appropriate option for the components being disassembled, as these components could be directly used again in other similar products after carrying out minor operations, such as cleaning, painting, lubricating, etc.; operations that involve low costs. The component could be sold in the market at the retail price.

- Concerning the REM option, components to be remanufactured require re-processing prior to reuse in new products. This involves manufacturing operations on the REM components, with higher costs than the REU option. The component could be sold in the market at the retail price.
- Regarding the REC alternative, it is a lower economic option than REU or REM, due to the components having to be recycled and because only raw materials could be removed from them. This option involves additional costs in the recycling operations. Only revenues obtained by selling these raw materials could be obtained from the recycled components.

First, the condition of the components is assumed to be known, as shown in Table 7, where the value of the $r_{i,m}$ indicator for the three considered approaches is supposed. Additionally, and due to the characteristics and functionality of some components, the disposal mode is considered for items 8, 14, 15, 16 and 17.

Table 7: Value of the recovery indicator ($r_{i,m}$) in the considered approaches

Item	**REU**				**REM**				**REC**			
n^{o}	$r_{i,1}$	$r_{i,2}$	$r_{i,3}$	$r_{i,4}$	$r_{i,1}$	$r_{i,2}$	$r_{i,3}$	$r_{i,4}$	$r_{i,1}$	$r_{i,2}$	$r_{i,3}$	$r_{i,4}$
1	0	0	1	0	0	0	1	0	0	0	1	0
2	0	0	1	0	0	0	1	0	0	0	1	0
3	0	0	1	0	0	0	1	0	0	0	1	0
4	0	0	1	0	0	0	1	0	0	0	1	0
5	0	0	1	0	0	0	1	0	0	0	1	0
6	0	0	1	0	0	0	1	0	0	0	1	0
7	1	0	0	0	0	1	0	0	0	0	1	0
8	0	0	0	1	0	0	0	1	0	0	0	1
9	1	0	0	0	0	1	0	0	0	0	1	0
10	1	0	0	0	0	1	0	0	0	0	1	0
11	1	0	0	0	0	1	0	0	0	0	1	0
12	1	0	0	0	0	1	0	0	0	0	1	0
13	1	0	0	0	0	1	0	0	0	0	1	0
14	0	0	0	1	0	0	0	1	0	0	0	1
15	0	0	0	1	0	0	0	1	0	0	0	1
16	0	0	0	1	0	0	0	1	0	0	0	1
17	0	0	0	1	0	0	0	1	0	0	0	1
18	1	0	0	0	0	1	0	0	0	0	1	0
19	1	0	0	0	0	1	0	0	0	0	1	0
20	1	0	0	0	0	1	0	0	0	0	1	0
21	0	0	1	0	0	0	1	0	0	0	1	0
22	0	0	1	0	0	0	1	0	0	0	1	0
23	0	0	1	0	0	0	1	0	0	0	1	0
24	0	0	1	0	0	0	1	0	0	0	1	0

The approaches under study were solved using the three calculation algorithms (CGA, HCA and GA). The results are presented in Figures 10, 11, and 12, and Table 8. These results generate findings and valuable insights, as follows:

- The REU approach is the most suitable alternative for the components in view of the better economic performance of the robotic disassembly process. The three algorithms propose a complete disassembly, with the best performance achieved by the GA (17.98 €), followed by HCA (16.70 €)

and CGA (15.52 €), as shown in Figure 10.

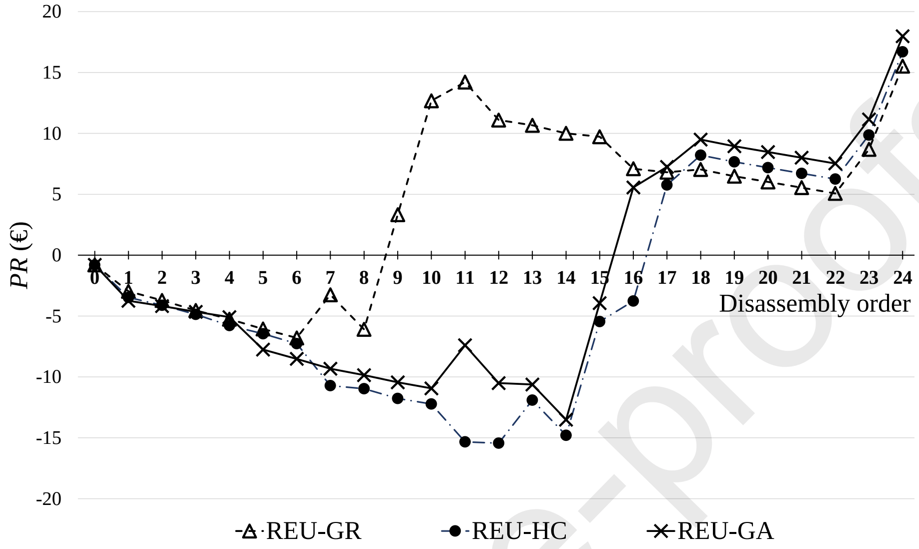


Figure 10: REU approach results. Economic profit *(PR)* in each -*th* position of the disassembly process for the three algorithms.

- From the analysis of the REM approach solution (Figure 11), the results reveal all three resolution algorithms propose a partial disassembly process. The GA suggests stopping the disassembly process after item 11 (Shaft A) is removed, obtaining a fitness value of 7.22 €. A similar disassembly sequence is proposed by CGA, with a fitness value of 6.78 €. As regards HCA, the disassembly process should be stopped after item 10 (Gear B) is removed.

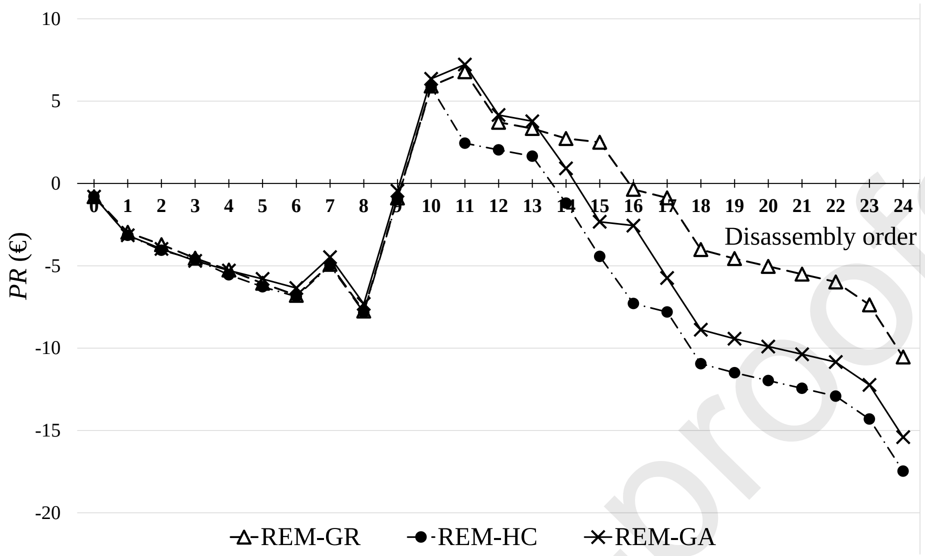


Figure 11: REM approach results. Economic profit *(PR)* in each *-th* position of the disassembly process for the three algorithms.

- Concerning the REC approach, from the analysis of Figure 12 and Table 8, the disassembly process is evidently not economically profitable as the robotic disassembly process costs are higher than the revenues from the recycling of the materials. Figure 12 shows the fitness value decreases in proportion to the disassembly level. Even so, and in the event that the complete disassembly is carried out, CGA proposes the most advantageous disassembly sequence, obtaining a fitness value of -31.64 €, followed by HCA with -41.00 €, and finally the GA with -49.29 €. The advantage of CGA over HCA and the GA in this case is the result of the type of evaluation used by each algorithm. CGA uses a local evaluation, selecting the best (in this case the one with less loss) at each step. In contrast, HCA and the GA use a global evaluation function, which is the value of using this sequence with its best threshold, and, in this case, the value is the same for all the sequences (permutations) because it obtains a threshold of 0. Therefore, the HCA and GA algorithms are searching in a plateau in which all the potential solutions (permutations) have the same fitness value, and thus, they can learn nothing from the fitness landscape. These results reveal valuable information for recyclers on the cost to be borne to perform a partial or a total disassembly process in the event that all components must be disposed of.

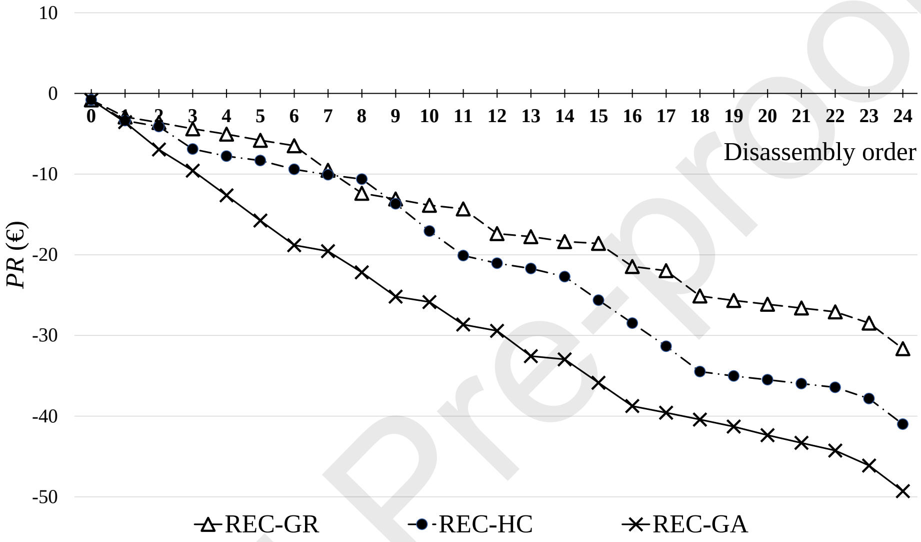


Figure 12: REC approach results. Economic profit *(PR)* in each *-th* position of the disassembly process for the three algorithms.

Table 8: Recovery approaches. Fitness value and disassembly sequence planning results.

Approach	Fitness value	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
REU-CGA	15.52	2	3	4	1	6	5	7	8	9	10	11	24	22	23	21	20	19	18	17	16	15	14	13
REU-HCA	16.70	4	2	3	5	1	6	23	24	21	22	19	20	7	8	10	11	9	18	17	16	15	14	13
REU-GA	17.98	23	21	24	22	2	1	6	5	4	3	7	19	20	8	9	10	11	18	17	16	15	14	13
REM-CGA	6.78	2	3	4	1	6	5	7	8	9	10	11	24	22	23	21	20	19	18	17	16	15	14	13
REM-HCA	5.84	3	1	2	6	5	4	7	8	9	10	23	24	22	11	21	20	19	18	17	16	15	14	13
REM-GA	7.22	3	4	5	6	1	2	7	8	9	10	11	24	22	20	23	21	19	18	17	16	15	14	13
REC-CGA	-31.64	2	3	4	1	6	5	7	8	9	10	11	24	22	23	21	20	19	18	17	16	15	14	13
REC-HCA	-41.00	24	23	3	5	2	4	1	6	7	21	8	10	9	19	22	11	20	18	17	16	15	14	13
REC-GA	-49.29	5	23	2	24	6	22	21	1	20	19	3	4	18	17	7	8	16	9	15	10	14	11	13

6.3. Sensitivity analysis

A sensitivity analysis is proposed in order to study how the evolution of certain variables subjected to uncertainty could modify the final results, forcing the process to be re-planned or decisions to be made about the final use of components. Figure 13 shows to what extent the cost per unit of time (c_t) determines the fitness value along the different proposed approaches. The 3D surface graph permits the evolution of the economic profit in the disassembly process to be analysed as a function of the c_t and the selected recovery approach: REU, REM or REC. According to the information from manufacturing companies in Spain, the c_t ranges from 120 €/hour (adopted as the lower specification limit, LSL) to 240 €/hour (adopted as the upper specification limit, USL).

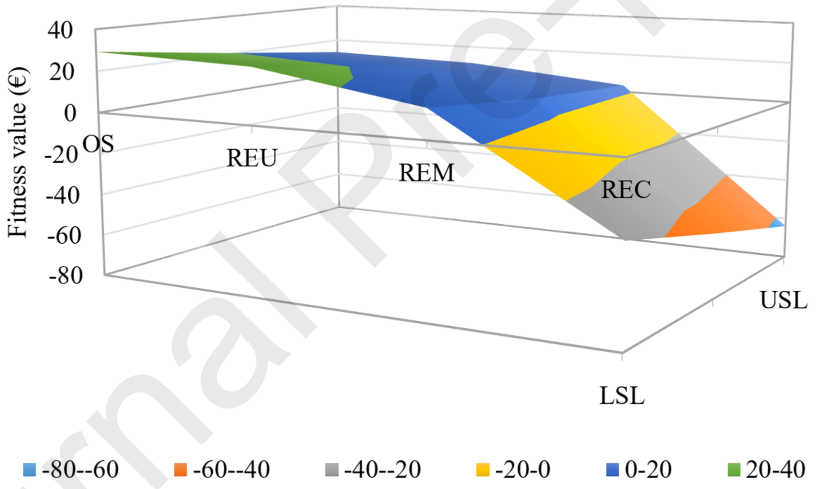


Figure 13: Sensitivity analysis of the cost per unit of time (c_T)

The calculations were performed using the GA, as this algorithm obtains a better performance and also in order to simplify the presentation of the results. For each c_t value and approach studied, the GA was run 100 times and the best one was selected to be shown in the graph. However, it is worth noting that the behaviour of the GA is highly stable, almost always converging to solutions with the same fitness.

The optimal solution (OS) obtains a range for the fitness value from 12.98 € in the USL to 29.29 € in the LSL. Concerning the REU approach, the fitness value ranges from 10.88 € to 26.71 €. In the case of the REM approach, the results range from 3.45 € to 13.98 €, and, finally, in the REC approach, the fitness value ranges from -33.72 € in the LSL to -63.95 € in the USL.

A similar operation was performed in the analysis of the retail price (RP_i), considering an evolution range of ±15 % according to the market prospect. The results are shown in Figure 14. Calculations were also performed with the GA and each simulation was repeated 100 times. The simulations produced the following results: the fitness value for the OS ranges from 10.80 € in the LSL to 32.02 € in the USL, whereas the REU approach solution ranges from 8.93 € to 28.63 €. The REM approach solution ranges from 2.21 € to 14.52 €, and, finally, the REC approach solution converges to -49.29 € as it does not consider the recovery of any component for reuse or remanufacturing.

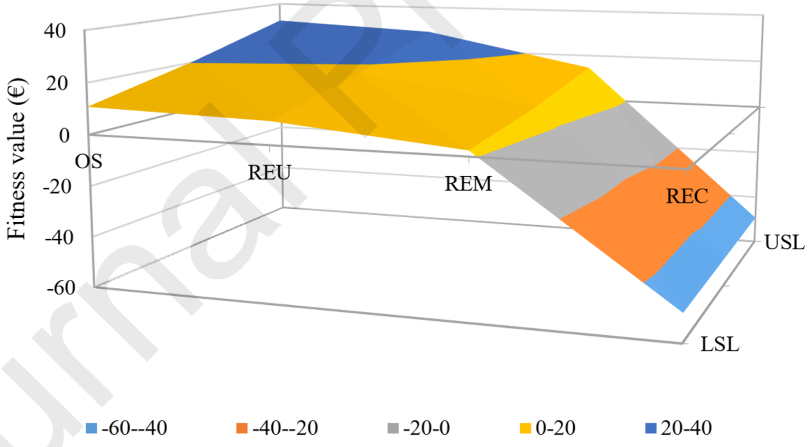


Figure 14: Sensitivity analysis of the retail price (RP_i)

7. Conclusions

This paper has presented a model for the economic assessment of the robotic disassembly process of end-of-life products. The contribution of this work is threefold. First, the model aims simultaneously to resolve the problems of disassembly sequence planning, the optimal disassembly level and selection of the recovery option for the components. Second, the model is able re-plan the optimal disassembly process level depending on the process uncertainties. Third, the model is applied to robotic disassembly. In the literature, most research works deal with disassembly sequence planning and recovery strategies as separate problems, but none of these studies has been applied to robotic disassembly or the re-planning of the disassembly strategy according to the product and process uncertainties.

The model takes into consideration all the parameters involved in the robotic process, such as disassembly tools, trajectories, penalties, costs, process times, constraints and others, in addition to the condition of components to be disassembled and their potential final use: reuse, remanufacturing, recycling or disposal. Due to the NP-complete nature of the problem, three algorithms based on heuristic are proposed to resolve the disassembly process: a Constructive Greedy Algorithm, a Hill Climbing Algorithm and a Genetic Algorithm. A case study based on a gear pump with 24 components is presented. Simulations were performed to test the proposed model showing its suitability to resolve the disassembly process, obtaining the most appropriate solution for the disassembly sequence planning, the optimal economic profit, the best recovery option for the disassembled components, and the stopping point of the process in the event that the solution proposes a partial disassembly process. In addition, the model is able to perform a sensitivity analysis of variables subjected to uncertain behaviour providing significant findings about how the process could be re-planned depending on the recovery approach.

Furthermore, our research has several implications for industrial activities. First, it can provide firms with a tool to improve their practices in recovery of EoL products, particularly gear pumps, although the work could also be applied to other products to be recovered. Second, the model provides firms with a methodology to assess the economic performance of the disassembly process allowing them to determine the optimal disassembly level of the product before the process is initiated or implemented. It is of great interest for companies from the perspective of the investments and resources required to put the disassembly process into effect. In addition, it is a robust tool that helps achieve economic goals. Third, the model allows decisions to be made regarding the ongoing disassembly process, hence increasing process flexibility, allowing the process to be paused before the predefined disassembly level, or to carry out more steps than those initially planned. All of this if the ongoing process performance, or the real state of the already disassembled components, requires making decisions about the process parameters or to reassign the recovery modes of the disassembled components, which are different to those initially planned. Fourth, the model provides firms with a tool to manage the recycling and disposal of components, in order to meet environmental objectives and to comply with legislation on recyclability.

Finally, as key factors in the success of this research, it is important to highlight the practical approach of the model, considering the need for the companies to manage these types of EoL products, and the support for remanufacturing companies, providing a pack of sound data to obtain a more realistic analysis of disassembly process performance.

Future work could focus on two areas. The first is the implementation of the model from the point of view of factory operation, in order to confirm the effectiveness of the model in an industrial scenario, validating the solutions provided by the algorithms with real data information obtained in the factory. Second is the consideration of the environmental and social dimensions of the robotic disassembly process so that together with the economic dimension analysed, the three dimensions of sustainability could be addressed as a multi-objective optimisation decision-making model.

Acknowledgements

[blind review]

Appendix

Notations and related descriptions for acronyms, parameters and subscripts are presented in Tables 9, 10 and 11.

Table 9: Acronyms

ABC	Artificial Bee colony
ACO	Ant Colony Optimisation
BA	Bees Algorithm
CAD	Computer aided design
CGA	Constructive Greedy Algorithm
DIS	Disposal
DLB	Disassembly line balancing
DPP	Disassembly path planning
DSP	Disassembly sequence planning
EoL	End-of-life
GA	Genetic Algorithm
HCA	Hill Climbing Algorithm
MRO	Material Recovery Opportunities
NP	Non-Deterministic Polynomial
OEM	Original Equipment Manufacturer
PR	Economic profit
PSO	Particle Swarm Optimisation
RDP	Robotic disassembly process
RDSP	Robotic disassembly sequence planning
REC	Recycling
REM	Remanufacturing
REU	Reuse

Table 10: Parameters

DS_G	Disassembly process gains
DS_C	Disassembly process costs
RP_i	Retail price
$r_{i,m}$	Indicator of the recovery mode: 1 if mode j is assigned to component i
RR_i	Revenue obtained from the component i to be recycled
CD_i	Disposal cost of the component i
C_{OP}	Total operation costs
C_{RC}	Total recovery costs
C_{OH}	Total overhead costs
C_{DP}	Total depreciation costs
t_T	Total process time
c_t	Cost per unit of time
$t_{p,q}$	Moving time between the adjacent disassembly operations x_p and x_q
$t_B(x_p, M, x_q)$	Moving time between disassembly operations and the tool magazine M
$t_{B1}(x_p, M)$	Moving time between the disassembly point of the operation x_p and the tool magazine M
x_p	Disassembly operation p
x_q	Disassembly operation q
$t_{B2}(M)$	Tool change time
$t_{B3}(M, x_q)$	Moving time between the tool magazine M and the disassembly point x_q
C	Precedence matrix
L	Length matrix
v_e	Linear velocity of the industrial robot's end-effect
$t_{B4}(x_p, M)$	Penalty time for process direction changes
P	Penalty time matrix
$t_{B5}(M, x_q)$	Penalty time for process direction changes along the path between the tool magazine M and x_q
$t_{C1}(x_p, x_q)$	Moving time between the disassembly points x_p and x_q
$t_{C2}(x_p, x_q)$	Penalty time for process direction changes along the path between x_p and x_q
$rc_{i,m}$	Recovering cost of the component i
$oh_{i,m}$	Overhead cost assigned to the component i
$dp_{i,m}$	Depreciation cost assigned to the component i
$P_{t,1}$	Penalty time if direction is changed by 90^o
$P_{t,2}$	Penalty time if direction is changed by 180^o
α_p	Indicator taking value 1 if the operation x_q must to be completed in the disassembly process, and 0 otherwise
$\delta_{p,q}$	Indicator taking value 1 if the operation x_q is carried out after the operation x_p, and 0 otherwise
α_i	Indicator that takes the value 1 if the component i is disassembled, and 0 otherwise
$\gamma_{p,q}$	Indicator taking the value 1 if the operation x_q requires the robot changes the tool in the tool magazine M

Table 11: Subscripts

i,j,k	Indicator of component
m	Indicator of recovery mode
p,q	Indicator of disassembly operation

Alshibli, M., El Sayed, A., Kongar, E., Sobh, T. M., Gupta, S. M., 2016. Disassembly sequencing using Tabu search. Journal of Intelligent & Robotic Systems 82 (1), 69–79.

Aydemir-Karadag, A., Turkbey, O., 2013. Multi-objective optimization of stochastic disassembly line balancing with station paralleling. Computers & Industrial Engineering 65 (3), 413–425.

Barwood, M., Li, J., Pringle, T., Rahimifard, S., 2015. Utilisation of reconfigurable recycling systems for improved material recovery from e-waste. Procedia CIRP 29, 746–751.

Büker, U., Drüe, S., Götze, N., Hartmann, G., Kalkreuter, B., Stemmer, R., Trapp, R., 2001. Vision-based control of an autonomous disassembly station. Robotics and Autonomous Systems 35 (3), 179–189.

Dorigo, M., Birattari, M., Stutzle, T., 2006. Ant Colony Optimization. IEEE Computational Intelligence Magazine 1 (4), 28–39.

ElSayed, A., Kongar, E., Gupta, S., Sobh, T., 2011. An online Genetic Algorithm for automated disassembly sequence generation. In: Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE. pp. 657–664.

Elsayed, A., Kongar, E., Gupta, S. M., 2010. A Genetic Algorithm approach to end-of-life disassembly sequencing for robotic disassembly. Proceedings of the 2010 Northeast Decision Sciences Institute Conference 1 (203), 402–408.

Elsayed, A., Kongar, E., Gupta, S. M., Sobh, T., 2012. A robotic-driven disassembly sequence generator for end-of-life electronic products. Journal of Intelligent and Robotic Systems: Theory and Applications 68 (1), 43–52.

Feldmann, K., Trautner, S., Lohrmann, H., Melzer, K., 2001. Computer-based product structure analysis for technical goods regarding optimal end-of-life strategies. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 215 (5), 683–693.

Gendreau, M., Potvin, J.-Y., 2010. Handbook of Metaheuristics. Vol. 2. Springer.

Ghandi, S., Masehian, E., 2015. Review and taxonomies of assembly and disassembly path planning problems and approaches. Computer-Aided Design 67, 58–86.

Ghoreishi, N., Jakiela, M. J., Nekouzadeh, A., 2013. A nongraphical method to determine the optimum disassembly plan in remanufacturing. Journal of Mechanical Design 135 (2), 021002.

Gil, P., Pomares, J., Diaz, S. v. P. C., Candelas, F., Torres, F., 2007. Flexible multi-sensorial system for automatic disassembly using cooperative robots. International Journal of Computer Integrated Manufacturing 20 (8), 757–772.

Go, T. F., Wahab, D. A., Rahman, M. N. A., Ramli, R., Hussain, A., 2012. Genetically optimised disassembly sequence for automotive component reuse. Expert Systems with Applications 39 (5), 5409–5417.

Gonçalves, J. F., de Magalhães Mendes, J. J., Resende, M. G., 2005. A hybrid genetic algorithm for the job shop scheduling problem. European journal of operational research 167 (1), 77–95.

Grabcad Community, 2018. Gear pump 10 l/min.
URL `https://grabcad.com/library/gear-pump-10l-min-1(Accessed: May22,2018)`

Gunantara, N., 2018. A review of multi-objective optimization: Methods and its applications. Cogent Engineering 5 (1), 1–16.

Gupta, S. M., Erbis, E., McGovern, S. M., 2004. Disassembly sequencing problem: a case study of a cell phone. In: Proc. SPIE. Vol. 5583. pp. 43–52.

Haj Darwish, A., Joukhadar, A., Kashkash, M., 2018. Using the Bees Algorithm for wheeled mobile robot path planning in an indoor dynamic environment. Cogent Engineering 5 (1), 1426539.

Hesselbach, J., 1994. Automatization in dismantling of printed circuit boards. RECY'94, 254–267.

Ilgin, M. A., Gupta, S. M., 2010. Comparison of economic benefits of sensor embedded products and conventional products in a multi-product disassembly line. Computers and Industrial Engineering 59 (4), 748–763.

Jin, G., Li, W., Wang, S., Gao, S., 2015. A systematic selective disassembly approach for waste electrical and electronic equipment with case study on liquid crystal display televisions. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 0954405415575476.

Jin, G., Li, W., Xia, K., 2013. Disassembly matrix for liquid crystal displays televisions. Procedia CIRP 11, 357–362.

Johnson, M. R., Wang, M. H., 1998. Economical evaluation of disassembly operations for recycling, remanufacturing and reuse. International Journal of Production Research 36 (12), 3227–3252.

Jovane, F., Semeraro, Q., Armillotta, A., 1998. On the Use of the Profit Rate Function in Disassembly Process Planning. The Engineering Economist 43 (4), 309–330.

Kalayci, C. B., Gupta, S. M., 2013. Artificial Bee Colony Algorithm for solving sequence-dependent disassembly line balancing problem. Expert Systems with Applications 40 (18), 7231–7241.

Karaboga, D., 2005. An idea based on Honey Bee Swarm for numerical optimization. Tech. rep., Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department.

Kongar, E., Gupta, S. M., 2006. Disassembly sequencing using Genetic Algorithm. International Journal of Advanced Manufacturing Technology 30 (5-6), 497–506.

KUKA, 2019. Technical Specification, LBR iiwa 7 R800, LBR iiwa 14 R820. Tech. rep., KUKA Deutschland GmbH.

Lambert, A. J., 2003. Disassembly sequencing: a survey. International Journal of Production Research 41 (16), 3721–3759.

Larranaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., Dizdarevic, S., 1999. Genetic algorithms for the travelling salesman problem: A review of representations and operators. Artificial Intelligence Review 13 (2), 129–170.

Li, J., Khoo, L., Tor, S., 2002. A novel representation scheme for disassembly sequence planning. The International Journal of Advanced Manufacturing Technology 20 (8), 621–630.

Liu, J., Zhou, Z., Pham, D. T., Xu, W., Ji, C., Liu, Q., 2018. Robotic disassembly sequence planning using Enhanced Discrete Bees Algorithm in remanufacturing. International Journal of Production Research 56 (9), 3134–3151.

Lourenço, H. R., Martin, O. C., Stützle, T., 2002. Iterated local search. In: Handbook of Metaheuristics, volume 57 of International Series in Operations Research and Management Science. Kluwer Academic Publishers, pp. 321–353.

Ma, Y.-S., Jun, H.-B., Kim, H.-W., Lee, D.-H., 2011. Disassembly process planning algorithms for end-of-life product recovery and environmentally conscious disposal. International Journal of Production Research 49 (23), 7007–7027.

Matsumoto, M., Ijomah, W., 2013. Remanufacturing. In: Handbook of Sustainable Engineering. Springer, pp. 389–408.

Meng, K., Lou, P., Peng, X., Prybutok, V., 2017. Multi-objective optimization decision-making of quality dependent product recovery for sustainability. International Journal of Production Economics 188 (October 2016), 72–85.

Michalewicz, Z., 1992. Genetic algorithms + data structures = evolution programs. Vol. 1. Springer-Verlag.

Michalewicz, Z., Fogel, D. B., 2013. How to solve it: modern heuristics. Springer Science & Business Media.

Miller, B. L., Goldberg, D. E., et al., 1995. Genetic algorithms, tournament selection, and the effects of noise. Complex systems 9 (3), 193–212.

Ondemir, O., Gupta, S. M., 2014. A multi-criteria decision making model for advanced repair-to-order and disassembly-to-order system. European Journal of Operational Research 233 (2), 408–419.

Percoco, G., Diella, M., 2013. Preliminary evaluation of Artificial Bee Colony Algorithm when applied to multi objective partial disassembly planning. Research Journal of Applied Sciences, Engineering and Technology 6 (17), 3234–3243.

Pham, D. T., Castellani, M., 2015. A comparative study of the Bees Algorithm as a tool for function optimisation. Cogent Engineering 2 (1), 1091540.

Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M., 2005. The Bees Algorithm. Technical Note, Manufacturing Engineering Centre, Cardiff University, UK.

Pham, D. T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., Zaidi, M., 2006. The Bees Algorithm—a novel tool for complex optimisation problems. In: Intelligent Production Machines and Systems. Elsevier, pp. 454–459.

Poli, R., Kennedy, J., Blackwell, T., 2007. Particle Swarm Optimization. Swarm Intelligence 1 (1), 33–57.

Ren, Y., Zhang, C., Zhao, F., Xiao, H., Tian, G., 2018. An asynchronous parallel disassembly planning based on genetic algorithm. European Journal of Operational Research 269 (2), 647–660.

Rickli, J. L., Camelio, J. A., 2013. Multi-objective partial disassembly optimization based on sequence feasibility. Journal of Manufacturing Systems 32 (1), 281–293.

Rickli, J. L., Camelio, J. A., 2014. Partial disassembly sequencing considering acquired end-of-life product age distributions. International Journal of Production Research 52 (24), 7496–7512.

Santochi, M., Dini, G., Failli, F., 2002. Computer aided disassembly planning: state of the art and perspectives. CIRP Annals-Manufacturing Technology 51 (2), 507–529.

Selman, B., Gomes, C. P., 2006. Hill-climbing search. Encyclopedia of Cognitive Science 81, 333–336.

Smith, S., Hsu, L.-Y., Smith, G. C., 2016. Partial disassembly sequence planning based on cost-benefit analysis. Journal of Cleaner Production 139, 729–739.

Song, X., Zhou, W., Pan, X., Feng, K., 2014. Disassembly sequence planning for electro-mechanical products under a partial destructive mode. Assembly Automation 34 (1), 106–114.

Torres, F., Puente, S., Díaz, C., 2009. Automatic cooperative disassembly robotic system: Task planner to distribute tasks among robots. Control Engineering Practice 17 (1), 112–121.

UNEP, 2017. United Nations Environmental Programme. Tech. rep., United Nations.
URL https://www.unep.org/es(accessed:22.06.2017)

Vongbunyong, S., Kara, S., Pagnucco, M., 2012. A Framework for Using Cognitive Robotics in Disassembly Automation Overview of Cognitive Robotics. Leveraging Technology for a Sustainable World Proceedings of the 19th CIRP Conference on Life Cycle Engineering, University of California at Berkeley, Berkeley, USA, May 23 - 25, 2012, 2–7.

Vongbunyong, S., Kara, S., Pagnucco, M., 2013. Application of cognitive robotics in disassembly of products. CIRP Annals - Manufacturing Technology 62 (1), 31–34.

Vongbunyong, S., Kara, S., Pagnucco, M., 2015. Learning and revision in cognitive robotics disassembly automation. Robotics and Computer-Integrated Manufacturing 34, 79–94.

Wang, H., Peng, Q., Zhang, J., Gu, P., 2017. Selective disassembly planning for the end-of-life product. Procedia CIRP 60, 512–517.

Yu, J.-M., Lee, D.-H., 2018. Scheduling algorithms for job-shop-type remanufacturing systems with component matching requirement. Computers & Industrial Engineering 120, 266–278.

Zhang, X.-h., Li, S.-q., Wang, J.-f., Li, X., 2007. Single object selective disassembly sequence planning based on Ant Colony Algorithm. Computer Integrated Manufacturing Systems - Beijing 13 (6), 1109.

Zhang, Z., Wang, K., Zhu, L., Wang, Y., 2017. A Pareto Improved Artificial Fish Swarm Algorithm for Solving a Multi-objective Fuzzy Disassembly Line Balancing Problem. Expert Systems with Applications.

Zhou, Z., Liu, J., Pham, D. T., Xu, W., Ramirez, F. J., Ji, C., Liu, Q., 2018. Disassembly sequence planning: Recent developments and future trends. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 0954405418789975.

HIGHLIGTHS

- Optimisation decision-making model in designing the robotic disassembly process
- Economic model for efficient and cost-effective use of components from EoL products
- Optimal disassembly level, disassembly plan, and recovery option solutions
- Re-planning of the optimal disassembly level and the recovery options for components
- Case study based on a gear pump to demonstrate the effectiveness of the model

CRediT author statement - MANUSCRIPT #CAIE-D-19-01132

Title: *Economic modelling of robotic disassembly in end-of-life product recovery for remanufacturing*

F. Javier Ramirez: Conceptualisation, Methodology, Validation, Formal analysis, Investigation, Writing - Original Draft, Writing - Review & Editing, Supervision

Juan A. Aledo: Methodology, Data Curation, Writing - Original Draft, Writing – Review & Editing

Jose A. Gamez: Software, Resources, Data Curation, Writing - Original Draft, Writing – Review & Editing

Duc T. Pham: Investigation, Data Curation, Writing – Review & Editing