A multiobjective decision-making approach for modelling and planning economically and environmentally sustainable robotic disassembly for remanufacturing

Hartono, Natalia; Ramirez, Francisco Javier; Pham, Duc

Computers & Industrial Engineering 184 (2023) 109535

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A multiobjective decision-making approach for modelling and planning economically and environmentally sustainable robotic disassembly for remanufacturing

Natalia Hartono [a,c,*], F. Javier Ramírez [b], D.T. Pham [a]

[a] Department of Mechanical Engineering, College of Engineering and Physical Sciences, The University of Birmingham, B15 2TT, UK
[b] School of Industrial Engineering, Department of Business Administration, Universidad de Castilla-La Mancha, 02071 Albacete, Spain
[c] Department of Industrial Engineering, University of Pelita Harapan, 15811, Indonesia

ARTICLE INFO

Keywords:
End-of-life
Remanufacturing
Disassembly sequence planning
Robotic disassembly
Multiobjective optimisation
Recovery option

ABSTRACT

Adopting initiatives to extend the useful life of products or recovering their components for reuse, remanufacturing or recycling is a key issue in the attempt to protect the environment and minimise the demand for natural resources. To contribute to the performance of automated disassembly practices, this paper presents a multiobjective decision-making approach based on the optimisation of three goals in a robotic disassembly cell framework: enhancing the economic performance of the process, reducing energy consumption and mitigating the environmental impact. Two real-use cases are presented as demonstrators, supported by appropriate, updated information from industry. The design model allowed the authors to obtain the best robotic disassembly sequence plan, the correct disassembly direction, the best recovery option for the disassembled components – reuse, remanufacturing, recycling or disposal – and the most appropriate disassembly tools, finding the optimal or near-optimal solution that best balances the three sustainability goals. An Enhanced Discrete Bees Algorithm with a mutation operator was employed to find the solution for the optimisation. Moreover, a multiobjective Bees Algorithm, a Non-dominated Sorting Genetic Algorithm II and a Pareto Envelope-based Selection Algorithm II were adopted to solve the multiobjective optimisation problem using different iteration numbers and population sizes. The results provide insights into robotic disassembly processes, encouraging firms to adopt more automated and sustainable remanufacturing strategies.

1. Introduction

The end of life (EoL) of a product is a burden for the environment if it is simply disposed of in a landfill site, polluting the air, soil and water. The enormous waste generated each year worldwide is a growing cause of concern among governments (UNEP, 2017), which are promoting initiatives and solutions to recover products and their components. The circular economy (CE) has emerged as a research trend to benefit the environment, economy and society, defined as "a regenerative system in which resource input and waste, emission, and energy leakage are minimised by slowing, closing, and narrowing material and energy loops…" (Geissdoerfer et al., 2017). EoL recovery options generally include part reuse, remanufacturing, material recycling and, as the last choice, disposal (Lee et al., 2010; Xia et al., 2014). These alternatives focus on saving raw materials, improving manufacturing costs, and reducing environmental impact.

Remanufacturing involves reducing waste, saving energy and optimising resource consumption in manufacturing (Chiodo and Ijomah, 2014), being one of the key sustainable alternatives to leverage resources and prolong the remaining useful life of products, while helping preserve the environment. Remanufacturing recreates a product using components from EoL products (Lambert and Gupta, 2004). Matsumoto and Ijomah (2013) defined remanufacturing as "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". Another definition of this term is formalised by the Remanufacturing Industries Council (RIC, 2017) as "a comprehensive and rigorous industrial process by which a previously sold, leased, used, worn, or nonfunctional product or part is returned to like-

* Corresponding author.
E-mail addresses: nxh886@student.bham.ac.uk (N. Hartono), franciscoj.ramirez@uclm.es (F.J. Ramírez), d.t.pham@bham.ac.uk (D.T. Pham).

https://doi.org/10.1016/j.cie.2023.109535
Received 1 July 2022; Received in revised form 21 June 2023; Accepted 10 August 2023
Available online 15 August 2023

new or better-than-new condition, from both a quality and performance perspective, through a controlled, reproducible, and sustainable process". Remanufacturing involves reducing waste, saving energy and optimising resource consumption in manufacturing (Chiodo and Ijomah, 2014), as one of the key sustainable alternatives to leverage resources and prolong the remaining useful life of products while helping preserve the environment.

The first and most crucial operation in remanufacturing is disassembly (Lambert, 2003; Wang et al., 2013, 2014; Xia et al., 2014; Zhou et al., 2018), which distinguishes remanufacturing from conventional manufacturing (Priyono et al., 2016). Disassembly is defined as the process of separating products into parts and/or subassemblies with necessary inspection and sorting operations, which is one of the critical points in effective and efficient product reprocessing (Han et al., 2013). Disassembly is not simply the reverse of the assembly process (Brennan et al., 1994; Fan et al., 2013; Touzanne et al., 2001), but has its own unique characteristics (Jovane et al., 1998; McGovern and Gupta, 2011). Disassembly sequence planning (DSP) consists of defining a detailed plan for removing components or subassemblies from complete products (Lambert, 2003; Zhou et al., 2018), involving the selection of the best sequence for the disassembly process, taking into account the preferences and restrictions between components and fasteners.

DSP is recognised in the literature as a nondeterministic polynomial time (NP)-complete and intractable problem, for whose resolution mathematical programming methods are unsuitable (Elsayed et al., 2010, 2012; Gonçalves et al., 2005). Earlier studies used exact methods for simple problems (Johnson and Wang, 1998; Lambert, 1999, 2006), but these methods are unable to solve more complex problems in which there is a greater number of components and the product to be disassembled is more complex. Recent studies have tended to use approximate algorithms based on metaheuristics given their ability to find a near-optimal solution in a reasonable computational time. Thus, in recent years, the use of these methods has grown more than others. In this sense, the literature includes significant works focused on solving the DSP as a single-objective problem using different techniques based on metaheuristics, such as the genetic algorithm (GA) (Chunming, 2016; Go et al., 2012; Gonnuru et al., 2013; Kheder et al., 2015; Tseng et al., 2018; Zhang et al., 2015), particle swarm optimisation (PSO) (Pornsing and Watanasungsuit, 2014; Zhong et al., 2011), ant colony optimisation (ACO) (Luo et al., 2016; Shan et al., 2007), artificial bee colony (ABC) (Percoco and Diella, 2013), the immune algorithm (IA) (Lu and Liu, 2012), simulated annealing (SA) (Azab et al., 2011), fruit fly optimisation (FFO) (Qu et al., 2015), the Bees Algorithm (BA) (Laili et al., 2019; Liu et al., 2018a, 2020b), and, more recently, the flatworm algorithm (FA) (Tseng et al., 2020). Additionally, metaheuristic techniques have been applied to solve DSP as a multiobjective problem, such as the nondominated sorting genetic algorithm (NSGA-II) (Meng et al., 2017; Shokohyar et al., 2014), the multiobjective GA (MOGA) (Hula et al., 2003; Rickli and Camelio, 2013, 2014), the multiobjective evolutionary algorithm (MOEA) (Jun et al., 2007), and the multiobjective Bees Algorithm (MOBA) (Xu et al., 2020).

Disassembly processes tend to be naturally complex, as they entail managing different types of uncertainties due to missing, corroded or worn-out parts and components in the products to be disassembled (Laili et al., 2019; Vongbunyong et al., 2013a). Most disassembly processes were traditionally conducted manually (Liu et al., 2020a,a; Ong, Chang, & Nee, 2021; Vongbunyong, Kara, & Pagnucco, 2013a), but automated disassembly using robots is increasingly being applied, as robots provide great efficiency and an increased ability to handle uncertainties in dynamic disassembly processes (Vongbunyong et al., 2013a). Additionally, hazardous and dangerous disassembly tasks are done more efficiently and safely by robots. A number of research works involving robotic disassembly can be found in the literature: Torres et al. (2009) used decision trees to distribute tasks between robots; Elsayed et al. (2010) proposed GA to solve robotic disassembly sequence planning (RDSP); Elsayed et al. (2012) used robot vision and GA for dismantling computers and minimising disassembly time; Vongbunyong et al. (2015) proposed a cognitive robotic system for low-cost disassembly automation to replace humans; Barwood et al. (2015) set up a lab-scale robotic cell to disassemble electronic control units from vehicles; Vongbunyong et al. (2017) transferred skills from humans to robots with cognitive ability to disassemble an LCD screen; Li et al. (2018) proposed an automated robotic approach for disassembly electric vehicles; Liu et al. (2018a) minimised disassembly time using an Enhanced Discrete Bees Algorithm (EDBA) to solve RDSP; Liu et al. (2020b) used an improved discrete BA to enhance the disassembly efficiency in a collaborative disassembly sequence planning and disassembly line balancing robotic process; and Ramírez et al. (2020) proposed a model for robotic disassembly that aimed to maximise profit.

Most of the above research is based on approaches to maximise profit, minimise process time, minimise cost, or a combination of these, while in some cases, also considering environmental issues, formulated as either single- or multiobjective problems. Previous studies in the RDSP were single-objective optimisation problems. Moreover, there are only three studies on RDSP on sustainability issues.

It is clear the management of EoL products is one of the main challenges to achieve the CE paradigm. Researchers and practitioners must work together to find solutions for recovering components, parts and raw materials from EoL products. This must be done using technical ways that allow balancing different goals at the same time, like economic, energy, environmental, etc. The development of methodologies and procedures to help practitioners perform disassembly operations using industrial processes based on sustainability goals, and make the most appropriate decisions about the final use of the disassembled components is essential to encourage companies to better manage the recovery of EoL products. So, the purpose of this study is to fill the gap in the area of robotic disassembly by proposing a multiobjective decision-making approach to find the optimal or near-optimal solution that maximises profit, energy savings and environmental benefits using a robotic cell framework and two real industrial case studies as demonstrators.

This paper makes several contributions to the literature. First, the proposed model determines a near-optimal solution for the problem, providing the disassembly sequence planning for the robotic process, the best recovery option for the components –reuse, remanufacturing, recycling or disposal–, the disassembly direction for all components, and the disassembly tools required to complete the disassembly operations. Second, this study is the first to provide insights to pinpoint how different iterations and population sizes (parameters used) can yield the same optimal results using statistical tests rather than visual graphical explanations. Third, the simulated solution approach closely replicates the real disassembly process, using real data and information from the use cases, describing the robotic cell framework and the disassembly tools, in addition to considering the feasible paths for the robot to move between disassembly points and with the tool magazine.

The rest of the paper is organised as follows. Section 2 reviews the literature and related work. The proposed model and methodology are explained in Section 3. Section 4 describes the case studies, the robotic cell, and the experiments. Section 5 is devoted to presenting and discussing the results. Finally, Section 6 concludes the paper.

2. Literature review

DSP is defined as a systematic approach to determine the best sequence of activity in the segregation of a product in its end of life (Dong and Arndt, 2003) in a detailed plan (Lambert, 2003). As described by Zhou et al. (2018), DSP comprises three steps: disassembly mode (complete or partial), disassembly modelling (disassembly precedence relationships), and disassembly planning methods (objective and optimisation method). The most popular disassembly model is graph-based, followed by matrix-based, Petri Net, and other methods (Zhou et al., 2018).

Table 1
Comparison of previous studies.

Study	Recovery Option				Decision Goals			Disassembly Output				Robotic operator	Decision making		Parameter Analysis	Optimisation method
	REU	REM	REC	DIS	Economic	Energy	Environmental	Sequence	Recovery	Direction	Tool		SO	MO		
Hula et al. (2003)			v	v	v	v		v	v					v		MOGA
Kongar and Gupta (2006)	v		v		v			v					v			GA-PPX
Jun et al. (2007)	v	v	v	v	v				v					v		MOEA
Shan et al. (2007)					v			v		v	v		v			ACO
Elsayed et al. (2010)	v		v	v	v			v				v	v			GA-PPX
Zhong et al. (2011)					v			v					v			PSO and Dijkstra
Azab et al. (2011)	v		v		v			v					v			SA
Ma et al. (2011)	v	v	v	v	v			v	v				v			Integer Programming
Agrawal and Pande (2011)					v			v		v			v		Visual	GA-PPX
Jun et al. (2012)	v	v	v	v	v				v				v			GA and Q-heuristic
Go et al. (2012)	v				v			v		v					Visual	GA
Elsayed et al. (2012)	v		v	v	v			v				v	v			GA-PPX
Lu and Liu (2012)					v			v					v		Visual	AIA
Vongbunyong et al. (2012, 2013a,b, 2015, 2017)												v				
Li et al. (2013)					v		v	v					v			PSO
Percoco and Diella (2013)	v		v	v	v		v	v								ABC
Rickli and Camelio (2013)		v	v		v		v	v						v	Visual	MOGA
Gonnuru et al. (2013)	v		v	v	v			v	v				v			GA
Xia et al. (2014)	v		v		v			v					v			TLBO
Shokohyar et al. (2014)	v	v		v	v		v		v					v		NSGA-II
Rickli and Camelio (2014)					v			v						v	Visual	MOGA
Ondemir and Gupta (2014a)	v	v	v	v	v		v	v	v					v		Linear Physical Programming
Ondemir and Gupta (2014b)	v	v	v	v	v		v	v	v					v		MILP
Johnson and McCarthy (2014)	v	v	v	v	v				v					v	Sensitivity Analysis	Integer Programming
Kheder et al. (2015)					v			v			v		v		Visual	GA-PPX
Qu et al. (2015)					v			v					v			FFO
Meng et al. (2016)	v	v	v		v			v	v				v			ICA
Luo et al. (2016)					v			v					v			ACO
Chunming (2016)					v			v					v			GA
Alshibli et al. (2016)	v		v	v	v							v	v			Tabu search
Meng et al. (2017)	v	v	v		v		v	v	v					v		NSGA-II
Tseng et al. (2018)					v			v		v	v		v			Block based GA
Liu et al. (2018a)					v			v		v	v	v	v		Visual	EDBA
Alshibli et al. (2018)	v		v	v				v	v			v	v			SA
Gao et al. (2018)								v		v	v	v	v			ABC
Laili et al. (2019)					v			v		v		v	v			BA
Tseng et al. (2020)					v			v		v	v		v			Flatworm Algorithm
Xu et al. (2020)					v			v		v		v		v	Visual	MOBA-Pareto
Fu et al. (2021)	v		v		v	v		v						v	Experiment Design	MOMVO
Gunji et al. (2021)			v	v	v	v	v	v	v				v			Stability graph cut-set
Wang et al. (2021)								v				v		v		MOABC, MOPSO, NSGA-II, SPEA2
This paper	v	v	v	v	v	v	v	v	v	v	v	v	v	v	Statistical	EDBA and MOBA-Pareto

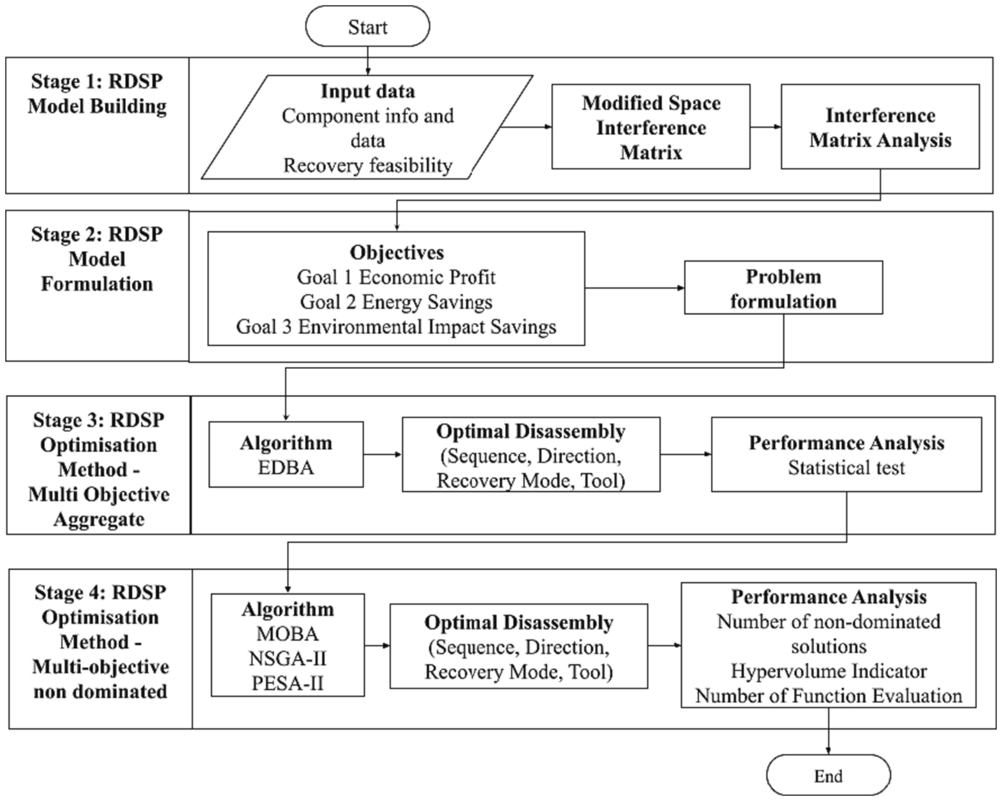


Fig. 1. RDSP optimisation decision-making model.

DSP belongs to the field of nondeterministic polynomial complete (NP) problems (Elsayed et al., 2012; Go et al., 2012; Huang et al., 2000; Kang and Xirouchakis, 2006; Meng et al., 2017; Tseng et al., 2018; Wang et al., 2013; Zhong et al., 2011). This means that to solve a complex case with an exact solution and using a consistent technique, high computational time is required because the number of possible sequences escalates exponentially with the number of components or parts (Go et al., 2012; Kuo, 2013; Tovey, 2002).

Most research on solving the DSP problem uses heuristic and metaheuristic mathematical programming (McGovern and Gupta, 2011). Zhou et al. (2018) reported that research mainly uses metaheuristic linear programming, rule-based methods and stochastic simulations to find the optimal disassembly sequence. Metaheuristic algorithms are known to be able to find a near-optimal solution in a reasonable computational time. Of the metaheuristic approaches, GA is the most frequently used to solve DSP, followed by PSO and ACO (Zhou et al., 2018). A small number of other metaheuristic algorithms have also been used to solve DSP, such as BA, IA, ABC, SA and FFO (Azab et al., 2011; Liu et al., 2018a; Lu and Liu, 2012; Percoco and Diella, 2013; Qu et al., 2015). To date, the most novel algorithm for solving DSP is FA, by Tseng et al. (2020). The BA (Pham et al., 2006) is a numerical optimisation technique inspired by honeybees' natural foraging behaviour. The BA has been successfully applied in neural network training, manufacturing cell formation, machine job shop scheduling, control system tuning, dynamic control problems, data clustering, mechanical design optimisation, image analysis, supply chain optimisation, robotic disassembly sequence and robotic disassembly line balancing (Laili et al., 2019; Liu et al., 2018a,b; Pham et al., 2006; Pham et al., 2014; Pham and Castellani, 2015; Yuce et al., 2013).

Concerning decision-making methods, single-objective (SO) and multiobjective (MO) approaches have mainly been used. Talbi (2009) suggests that a robust optimisation must find a trade-off between the quality of the solutions and the robustness of the solutions if there is interference in the decision variables, defined as MO optimisation. The complexity of the MO problem increases as the size of the problem escalates (Talbi, 2009). The MO approach provides a set of solutions, known as the Pareto Optimal Solutions (POSs), when it is impossible to enhance one of the goals without degenerating at least one of the others (Talbi, 2009).

The performance evaluation for MO optimisation is more complicated than the SO because the output is not a single solution but a set of solutions (Halim et al., 2020; Talbi, 2009). Solution-quality indicators for MO optimisation are divided into convergence-based indicators, diversity-based indicators and hybrid indicators (Talbi, 2009). The number of non-dominated solutions is one of the most widely used indicators to measure convergence speed, as proposed by Durillo et al. (2010). The diversity indicator measures the dispersion and extension of the solutions (Talbi, 2009), while the Hypervolume Indicator (HI) is a hybrid-based indicator to measure convergence and diversity (Cao et al., 2015; Talbi, 2009). HI is the most popular method (Zitzler and Thiele, 1998), and has become a standard indicator to measure the performance of MO optimisation algorithms (Zitzler et al., 2008). A higher HI is desirable because it shows a wider range of POSs (Halim et al., 2020). Another measurement of speed to compare the performance of algorithms is the Number of Function Evaluation (NFE) (Halim et al., 2020). Unlike the computational time, which has the major drawback of depending on computer characteristics, NFE is an independent measurement from the computer system (Talbi, 2009), and a reliable measurement for computation complexity (Halim et al., 2020). Most researchers use NFE to compare algorithms in SO problems, while NFE is used to compare algorithms in MO problems (Deb and Jain, 2013; Sun et al., 2018).

Solving the RDSP problem using BA was first presented by Liu et al. (2018a). In this work, an enhanced discrete bees algorithm (EDBA) using a mutation operator and EDBA without a mutation operator (EDBA-WMO) was proposed. The results show that EDBA using the mutation operator better minimises the total disassembly time compared to EDBA-WMO, the genetic algorithm with precedence preserve crossover (GA-PPX) and self-adaptive simplified swarm optimisation (SASSO). Studies of RDSP using BA have been conducted by Laili et al. (2019, 2022), Liu et al. (2018a), Chen et al. (2020) and Hartono et al. (2022a,b,c). Those studies primarily focus on single-objective optimisation, with the main objective being the minimisation of disassembly time, except for Hartono et al. who explored different objectives. This research notably introduces the first application of a multiobjective Bees Algorithm (MOBA) to the autonomous identification of the optimal recovery strategy in RDSP. An explanation of the differences between the single-objective and multiobjective versions of the BA, along with the specific versions utilised in this study, will be provided in the next section. Within this study, the MOBA is employed to identify the optimal recovery strategy for each disassembly component, representing a novel aspect that has not been previously addressed in the literature. By leveraging its inherent decision-making capabilities, the MOBA dynamically evaluates and selects the most favourable recovery options for each component, taking into account the specified constraints and adeptly balancing the objectives.

To better explain the contributions of this study and the research gap it aims to fill, Table 1 summarises and compares related previous research in DSP. Most of the previous research in DSP has proposed SO approaches, with there being a lack of MO optimisation techniques. Only six research works (Johnson and McCarthy, 2014; Jun et al., 2007, 2012; Ma et al., 2011; Ondemir and Gupta, 2014a,b) consider the EoL recovery option as reuse, remanufacturing, recycling and disposal, using human/manual disassembly. RDSP using an MO approach with the Bees Algorithm as the optimisation method is presented in the work by Xu et al. (2020), but EoL recovery options are not considered. Only three studies have sustainability as a goal (Alshibli et al., 2018; Gao et al., 2018; Wang et al., 2021). Of the types of goals considered in previous work, economic ones are the most commonly addressed. Our research, however, explores energy savings and environmental benefits as objectives to achieve sustainability, in addition to maximising profit.

3. Model and methodology

The RDSP decision-making model designed in this research consists of four stages: model building, model formulation, optimisation method-MO aggregate approach, and optimisation method-MO non-dominated approach. Fig. 1 shows a general overview of the proposed model, and the different stages are explained in the following sections.

3.1. Stage 1: RDSP model building

The first stage of the model is devoted to evaluating the interference between components and the precedence relationships between them to eliminate infeasible sequences (Zhou et al., 2018). At the beginning, the complete information on the products, components, properties and their recovery feasibility is collected. To this end, the model is provided with the information from the CAD design. Additionally, as opposed to manual disassembly, the model must be informed of the disassembly direction to define the suitable paths for the robot movements. Subsequently, feasible disassembly sequences and directions are generated using modified space interference matrix and interference matrix analyses, also called modified feasible solution generation (MFSG) (Liu et al., 2018a). Space interference matrix and interference matrix analyses were proposed by Jin et al. (2013, 2015) to represent disassembly precedence between components in six directions (X+, X-, Y+, Y-, Z+, Z-). The interference matrix M can be written as Eq. (1).

$$M = \begin{bmatrix} 0 & M_{12} & \cdots & M_{1n} \\ M_{21} & 0 & \ldots & M_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n1} & M_{n2} & \cdots & 0 \end{bmatrix} \tag{1}$$

where each element in the matrix is a multidimensional vector that represents whether part j is blocking the movement of part i along X+, X-, Y+, Y-, Z+ or Z- direction. If it is blocking, M_{ij} is 1, otherwise it is 0. Jin et al. (2013, 2015) defined the matrix for the negative direction as the transpose matrix of the positive direction. For example, matrix M for X- is the transpose matrix of X +. However, for case studies including bolts as fasteners, this methodology is unsuitable because, in the transpose matrix, the components fastened with bolts could be disassembled before the bolts, which it is not feasible. Therefore, this work considers the MFSG approach as proposed by Liu et al. (2018a), in which each matrix is separately evaluated to avoid this issue. It was also selected for its practicality in the disassembly process.

3.2. Stage 2: RDSP model formulation

Resolving the robotic disassembly process is proposed as a multi-objective problem with the aim of identifying an optimal or near-optimal solution (OS) to best balance the three goals: profit (f_1), energy savings (f_2) and environmental benefits (f_3), as shown in Eq. (2):

$$OS = \max(f_1, f_2, f_3) \tag{2}$$

The three goals are assessed according to the disassembly process outcomes and the subsequent recovery and sale in the market of the components that have been disassembled, depending on the recovery option selected for each of them: reuse, remanufacturing, recycling, or disposal.

3.2.1. Goal 1. Profit

f_1 is defined as shown in Eq. (3). The equation includes seven main addends: the total revenue obtained from the components to be reused or remanufactured, the total revenue obtained from the components to be recycled, the total disposal costs of the components to be disposed of, the total disassembly cost, the total recovery costs of the components to be reused or remanufactured, the overhead costs of the company and the depreciation costs of the machinery (robotic cell) used in the disassembly process.

$$\begin{aligned} f_1 = & \sum_{i=1}^{N}\sum_{j=1}^{2} RP_i r_{i,j}\alpha_i + \sum_{i=1}^{N} RC_i r_{i,3}\alpha_i - \sum_{i=1}^{N} CD_i r_{i,4}(1-\alpha_i) \\ & - \left[\sum_{i=1}^{N-1} t_b(x_i)\alpha_i c_T + \sum_{i=1}^{N-1}\left(\frac{PD(x_i, M)}{v_e} + t_c(x_i, x_{i+1}) + \frac{PD(M, x_{i+1})}{v_e}\right.\right. \\ & \left. + t_u(x_i, M) + t_w(M, x_{i+1})\right)\gamma_i\alpha_i c_T + \sum_{i=1}^{N-1}\left(\frac{PD(x_i, x_{i+1})}{v_e}\right. \\ & \left.\left. + t_z(x_i, x_{i+1})\right)(1-\gamma_i)\alpha_i c_T\right] - \sum_{i=1}^{N}\sum_{j=1}^{2} rc_{i,j} r_{i,j}\alpha_i - \sum_{i=1}^{N}\sum_{j=1}^{4} oh_{i,j} r_{i,j}\alpha_i \\ & - \sum_{i=1}^{N}\sum_{j=1}^{4} dp_{i,j} r_{i,j}\alpha_i \end{aligned} \tag{3}$$

where:

- i is the index for each component and varies from 0 to N
- j is the indicator of the recovery mode and equal to 1 if component i is assigned to be reused, 2 if it is to be remanufactured, 3 if it is to be recycled or 4 if it is to be disposed of.
- RP_i is the revenue obtained due to the component i to be reused or remanufactured not having been manufactured again for a new product
- $r_{i,j}$ is an indicator of the recovery mode: 1 if mode j is assigned to component i
- α_i is an indicator that takes the value of 1 if component i is to be disassembled and 0 otherwise.
- RC_i is the revenue obtained from component i being recycled
- CD_i is the disposal cost of component i being disposed of

- $t_b(x_i)$ is the basic time to perform disassembly operation x_i
- c_T is the cost per unit of time
- $PD(x_i,M)$ is the distance between the point of the disassembly operation x_i and the position of the tool magazine (M)
- v_e is the line velocity of the industrial robot's end effector
- $t_c(x_i,x_{i+1})$ is the tool change time and depends on the tool type
- $PD(M,x_{i+1})$ is the length between the position of the tool magazine (M) and the point of the disassembly operation x_{i+1}
- $t_u(x_i,M)$ is the penalty time for process direction changes along the path between x_i and the tool magazine (M) and formulated as follows:
  - 0 if the direction is not changed.
  - p_1 if the direction is changed by 90°.
  - p_2 if the direction is changed by 180°.
- $t_w(M,x_{i+1})$ is the penalty time for process direction changes along the path between the tool magazine (M) and x_{i+1}, and is formulated as t_u
- γ_i is an indicator taking the value 1 if operation x_{i+1} requires changing the tool used in previous operation x_i
- $PD(x_i,x_{i+1})$ is the distance between the point of the disassembly operation x_i and the point of disassembly operation x_{i+1}
- $t_z(x_i,x_{i+1})$ is the penalty time for process direction changes along the path between x_i and x_{i+1}, and formulated as t_u
- $rc_{i,j}$ is the recovery cost of component i being reused or remanufactured
- $oh_{i,j}$ is the overhead cost assigned to component i to be disassembled
- $dp_{i,j}$ is the depreciation cost assigned to component i to be disassembled

3.2.2. Goal 2. Energy savings

f_2 represents the energy savings obtained as a result of the disassembly process and the subsequent recovery of the components. As some of the disassembled components will be reused or remanufactured, the model considers the energy savings obtained due to these components not having been manufactured again in the production of new products. Eq. (4) shows the formulation of this goal, considering four main addends: the total reclaimed energy from the components to be reused or remanufactured, the total energy consumption of the robot in the overall disassembly process, the total energy consumption involved in recovering the components to be reused, remanufactured or recycled, and the total energy consumption in the final treatment of the components to be disposed of.

$$\begin{aligned} f_2 = & \sum_{i=1}^{N}\sum_{j=1}^{2} r_{i,j}\mathrm{gr}_{i,j}f_w\alpha_i - \sum_{i=1}^{N-1}\left[t_b(x_i)\mathrm{PR}_1\gamma_i + \frac{PD(M,x_i)\mathrm{PR}_2\gamma_i}{v_e}\right. \\ & + t_c(x_i,x_{i+1})\mathrm{PR}_2\gamma_i + \frac{PD(M,x_{i+1})\mathrm{PR}_2\gamma_i}{v_e} \\ & \left. + \frac{PD(x_i,x_{i+1})\mathrm{PR}_2(1-\gamma_i)}{v_e}\right]\frac{f_w\alpha_i}{3600} - \sum_{i=1}^{N}\sum_{j=1}^{3} r_{i,j}\mathrm{gc}_{i,j}f_w\alpha_i \\ & - \sum_{i=1}^{N} r_{i,4}\mathrm{gc}_{i,4}f_w(1-\alpha_i) \end{aligned} \tag{4}$$

where, in addition to the variables and parameters defined in the formulation of f_1, the following are considered:

- $gr_{i,j}$ is the energy reclaimed from component i being reused or remanufactured
- f_W is a conversion factor from kWh to monetary units
- $gd_{1,i}(x_i)$ is the energy consumption of the robot in the disassembly operation of component i
- $gd_{2,i}(x_i,M)$ is the energy consumption of the robot in the movement between the position x_i and M
- $gd_{3,i}(M)$ is the energy consumption of the robot in the tool change
- $gd_{4,i}(M,x_{i+1})$ is the energy consumption of the robot in the movement between M and x_{i+1}
- $gd_{5,i}(x_i,x_{i+1})$ is the energy consumption of the robot in the movement between x_i and x_{i+1}
- $gc_{i,j}$ is the energy consumption involved in recovering component i with mode j
- PR_1 is the power of the robot used in the disassembly operation
- PR_2 is the power of the robot used in the movements between the disassembly points

3.2.3. Goal 3. Environmental benefits

The third goal f_3, assesses the environmental benefits reclaimed in the disassembly process and the subsequent recovery of components. As shown in Eq. (5), five addends are considered in the formulation of this goal: the total reclaimed environmental benefits from the components to be reused or remanufactured; the total environmental benefits caused by the process of recovering components to be reused, remanufactured or recycled; the total environmental benefits incurred in the treatment of components to be disposed of; the total environmental benefits incurred in the disassembly operations; and the total environmental benefits produced in the movements of the robot between the disassembly points.

$$\begin{aligned} f_3 = & \sum_{i=1}^{N}\sum_{j=1}^{2} r_{i,j}er_{i,j}\alpha_i - \sum_{i=1}^{N} \\ & \times \sum_{j=1}^{3} r_{i,j}ec_{i,j}\alpha_i - \sum_{i=1}^{N} r_{i,4}ec_{i,4}(1-\alpha_i) - \sum_{i=1}^{N-1} ed(x_i)\alpha_i - \sum_{i=1}^{N-1} ed(x_i,x_{i+1})\alpha_i \end{aligned} \tag{5}$$

where, in addition to the variables and parameters presented in the definition of f_1 and f_2, the following are defined:

- $er_{i,j}$ is the reclaimed environmental benefits from component i being reused or remanufactured
- $ec_{i,j}$ is the environmental benefits in the recovering process of component i with mode j
- $ed(x_i)$ represents the environmental benefits in disassembly operation x_i.
- $ed(x_i,x_{i+1})$ represents the environmental benefits produced by the movement of the robot between disassembly operations x_i and x_{i+1}, considering that the robot has to change the tool in M if operation x_{i+1} requires using a different tool to the one used in the previous operation x_i.

3.2.4. Constraints

Finally, some constraints must be considered to complete the model formulation, as shown in Eqs.(6) to (9):

$$\sum_{j=1}^{4} r_{i,j} = 1 \qquad \forall i \tag{6}$$

$$r_{i,1} + r_{i,2} + r_{i,3} \leq \alpha_i \tag{7}$$

$$\alpha_i \geq \alpha_{i+1} \tag{8}$$

$$\sum_{i=1}^{N} \alpha_i \leq N-1 \tag{9}$$

where

- Eq. (6) guarantees that each component, i, has only one recovery mode.
- Eq. (7) assures that all components to be reused, remanufactured or recycled must be disassembled.
- Eq. (8) guarantees that if the disassembly operation of component i is the prerequisite of the disassembly operation of component $i+1$, component i must be disassembled.
- Eq. (9) guarantees the maximum number of total disassembled components.

```
Algorithm 1: The pseudo-code of EDBA for RDSP
Input: n: number of scout bees, m: number of best sites,  e: number of elite sites, nsp:
       recruited bees for best sites, nep: recruited bees for elite sites,  dis_m: robotic
       disassembly information matrix
Output: RDSP(sequence, direction, mode, tool, (f1, f2, f3  or  *f = sum(f1,f2,f3))
        *MOBA aggregate approach
Function EDBA(n, m, e, nsp, nep):
  Start
  // Generate initial population  with feasible disassembly sequence
  initialRDSP ← GlobalMFSG(dis_m : sequence, direction, mode)
  while stopping criterion not met do
    Evaluate population fitness
    f ← FVALUE(initialRDSP)
    Sort population according to f
    Select m best RDSP for local search
    // Generate local RDSP with waggle dance
    for each bee ∈  e do
      for each bee ∈  nep do
        // Do feasibility check
        while feasibility not met do do
          LocalBee ← WaggleDance(dis_m : sequence, direction)
        // Mutate the disassembly direction and recovery while keep the
         sequence
        LocalBee ← Mutation(dis_m : direction, mode)
        Evaluate fitness of Local_Bee
        if Local_Bee  better  than  Best_Bee  then
          // Update Best Bee
          Best_Bee  =  Local_Bee
    for each bee ∈  m − e do
      for each bee ∈  nsp do
        // Do feasibility check
        while feasibility not met do do
          LocalBee ← WaggleDance(dis_m : sequence, direction)
        // Mutate the disassembly direction and recovery while keep the
         sequence
        LocalBee ← Mutation(dis_m : direction, mode)
        Evaluate fitness of Local_Bee
        if Local_Bee  better  than  Best_Bee  then
          Best_Bee  =  Local_Bee
    // Assign remaining bees for global search
    for each bee ∈  n − m do
      global_Bee ← GlobalRDSP(dis_m : sequence, direction, mode)
    Evaluate fitness of the new population
    Sort population according to f
    Best RDSP = Best Bee (Bee with maximum f)
  return Best RDSP (Best Bee)
```

Fig. 2. EDBA pseudocode.

ILUSTRATION OF:

SWAP

Disassembly sequence	1	2	3	4	5
Disassembly direction	+X	+Y	-X	-Z	-Y
Disassembly recovery mode	1	4	2	3	1
Disassembly tool	Tool 1	Tool 2	Tool 1	Tool 3	Tool 2

Disassembly sequence	1	5	3	4	2
Disassembly direction	+X	-Y	-X	-Z	+Y
Disassembly recovery mode	1	1	2	3	4
Disassembly tool	Tool 1	Tool 2	Tool 1	Tool 3	Tool 2

INSERT

1	2	3	4	5
+X	+Y	-X	-Z	-Y
1	4	2	3	1
Tool 1	Tool 2	Tool 1	Tool 3	Tool 2

1	4	2	3	5
+X	-Z	+Y	-X	-Y
1	3	4	2	1
Tool 1	Tool 3	Tool 2	Tool 1	Tool 2

MUTATION

1	2	3	4	5
+X	+Y	-X	-Z	-Y
1	4	2	3	1
Tool 1	Tool 2	Tool 1	Tool 3	Tool 2

1	2	3	4	5
-X	+Y	-X	-Z	-Y
4	4	2	3	1
Tool 1	Tool 2	Tool 1	Tool 3	Tool 2

Fig. 3. Illustration of neighbourhood search in this paper.

3.3. Stage 3: RDSP optimisation method - multiobjective aggregate approach

At this stage, the RDSP model performs multiobjective aggregate approaches optimisation to find an optimal or near-optimal solution for each goal, as described in the previous section. The aggregate approach sums all the goals and treats them as a single objective. EDBA, an improvement of the BA using a mutation operator as proposed by Liu et al. (2018a), is adapted to find suitable solutions applying different iteration and population sizes to test the performance of the algorithm. The pseudocode of EDBA is depicted in Fig. 2.

The first step of the algorithm is focused on the parameter setting and the application of the stopping criteria (maximum iteration number). A number of scoutbees n are generated using MFSG for all feasible

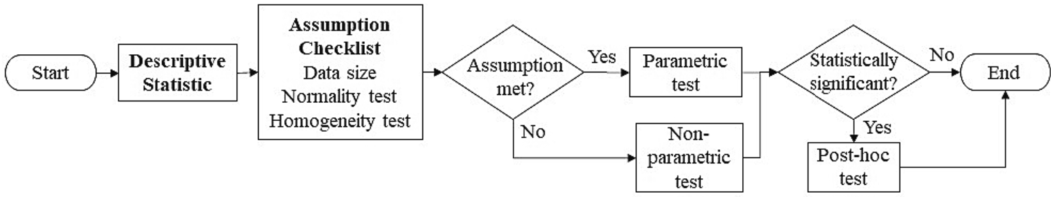


Fig. 4. Statistical test decision flowchart.

```
Algorithm 2: The pseudo-code of MOBA for RDSP
Input: n: number of scout bees, m: number of best sites, e: number of elite sites, nsp:
       recruited bees for best sites, nep: recruited bees for elite sites, dis_m: robotic
       disassembly information matrix
Output: RDSP(sequence, direction, mode, tool, POSs)
Function MOBA(n, m, e, nsp, nep):
    Start
    // Generate initial population  with feasible disassembly sequence and pareto
       front set
    initialRDSP ← GlobalMFSG(dis_m : sequence, direction, mode)
    while stopping criterion not met do
        Evaluate population fitness
        f ← FVALUE(initialRDSP)
        Sort population based on non-dominated sorting
        Select m best RDSP for local search
        // Generate local RDSP with waggle dance
        for each bee ∈ e do
            for each bee ∈ nep do
                // Do feasibility check
                while feasibility not met do do
                    LocalBee ← WaggleDance(dis_m : sequence, direction)
                // Mutate the disassembly direction and recovery while keep the
                   sequence
                LocalBee ← Mutation(dis_m : direction, mode)
                Evaluate fitness of Local_Bee based on non-dominated sorting
                if Local_Bee better than Best_Bee then
                    // Update Best Bee
                    Best_Bee = Local_Bee
        for each bee ∈ m − e do
            for each bee ∈ nsp do
                // Do feasibility check
                while feasibility not met do do
                    LocalBee ← WaggleDance(dis_m : sequence, direction)
                // Mutate the disassembly direction and recovery while keep the
                   sequence
                LocalBee ← Mutation(dis_m : direction, mode)
                Evaluate fitness of Local_Bee based on non-dominated sorting
                if Local_Bee better than Best_Bee then
                    Best_Bee = Local_Bee
        // Assign remaining bees for global search
        for each bee ∈ n − m do
            global_Bee ← GlobalRDSP(dis_m : sequence, direction, mode)
        Evaluate fitness of the new population
        Sort population based on non-dominated sorting
        // Store the Pareto frontier
        Best RDSP = Best Bee
    return Best RDSP (Best Bee)
```

Fig. 5. MOBA pseudocode.

disassembly sequences. Then, *n* is sorted by the fitness value. The best of *n* are elite site bees, *nep*, and search in the elite sites, *e*, using a neighbourhood strategy (swap, insert and mutation). The swap and insert operators move the disassembly sequence, direction, recovery mode and tools, whereas the mutation only changes the direction and the recovery mode, as depicted in Fig. 3. The mutation operator mutates the best bee of the *nep* to find the best fitness value. If the fitness value of the mutated bees is higher, then the best bee of *nep* is replaced and not changed otherwise. The process for selected sites *m* is similar to that with the elite sites *e*. The remaining bees $n - m$ perform a random search using MFSG to explore the solution space. The population is sorted by fitness value and updates the best RDSP information until the maximum number of iterations is reached.

Subsequently, a statistical test is carried out to find the number of

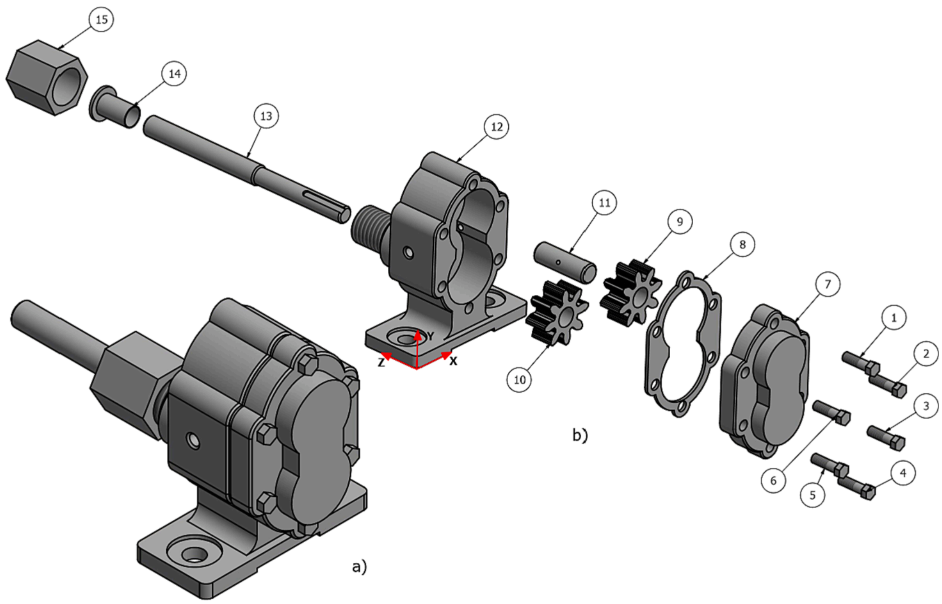


Fig. 6. Gear pump A: (a) assembled view; (b) exploded view. Source: Liu et al. (2018a).

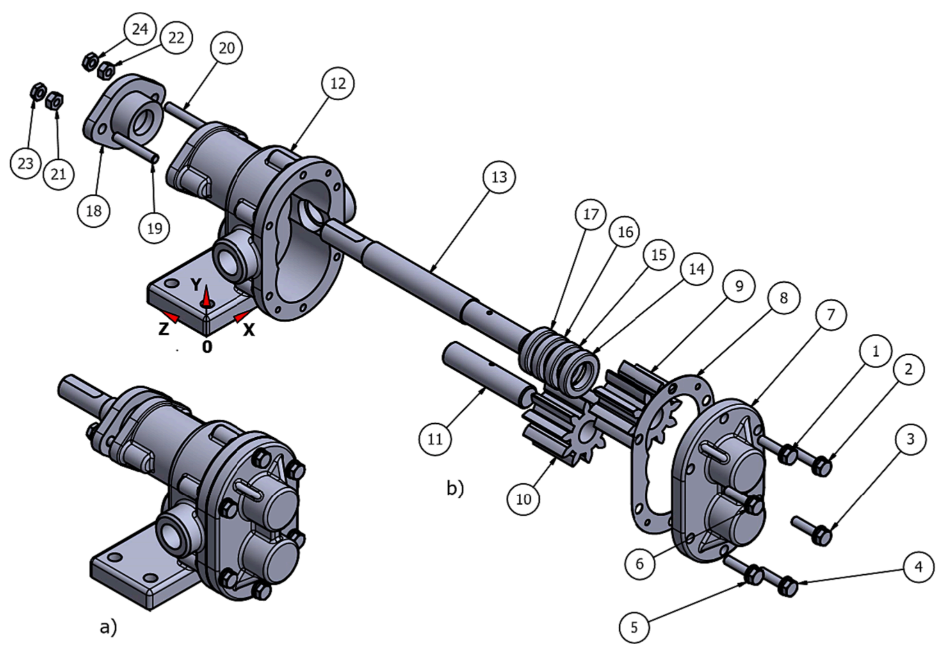


Fig. 7. Gear pump B: (a) assembled view; (b) exploded view. Source: Grabcad Community (2020) and Ramírez et al. (2020).

iteration and population sizes that result in the maximum fitness values. Fig. 4 shows the decision flowchart for the proposed statistical test. First, a descriptive test generates a boxplot graph to visually show the data distribution, median and outliers. A pretest is then performed to check the assumptions required to satisfy the parametric test: (i) the number of samples is higher than 30, (ii) the data fit a normal distribution, and (iii) the data belong to the same population. The parametric test is only carried out if the three assumptions are met. Otherwise, the nonparametric test will be performed. The next step is devoted to determining whether there are significant differences between the groups. If there are no significant differences, then any iteration and population size could be used to find the maximum fitness value. However, if the statistical test shows that the independent groups have significant differences, a post hoc test is performed with pairwise multiple comparisons to determine which groups are significantly different from other groups. The results will pinpoint which groups can be used to find the maximum fitness value.

3.4. Stage 4: RDSP optimisation method – multiobjective non dominated approach

In the last stage, MO optimisation is applied to find the best balance of the three goals. As a result, a set of POSs and the image of this set in an objective space called the Pareto front (Talbi, 2009) is obtained. The POSs will help the decision-maker adopt the most appropriate choice for their preferences.

This research proposes two approaches to solve MO problems. The first is a scalar approach using the aggregation method. This approach transforms the MO problem into an SO problem by linearly adding the objective function (Talbi, 2009). This first approach uses EDBA, since it

Table 2
Gear pump A. Properties and disassembly requirements for all components.

Item	Name	Material	Volume (mm^3)	Weight (g.)	Disassembly point X	Y	Z	Disassembly tool	$t_b(x_i)$ (s)
1	Bolt A	Steel	1,006.5	7.9	49.4	105.5	−12.6	Spanner-I	3
2	Bolt B	Steel	1,006.5	7.9	74.4	81	−12.6	Spanner-I	3
3	Bolt C	Steel	1,006.5	7.9	74.4	45	−12.6	Spanner-I	3
4	Bolt D	Steel	1,006.5	7.9	49.4	20.5	−12.6	Spanner-I	3
5	Bolt E	Steel	1,006.5	7.9	24.4	45	−12.6	Spanner-I	3
6	Bolt F	Steel	1,006.5	7.9	24.4	81	−12.6	Spanner-I	3
7	Cover	Steel	68,552.5	538.1	49.4	63	−20.6	Gripper-II	4
8	Gasket	Rubber	4,450.4	4.2	49.4	105.5	1.4	Gripper-I	3
9	Gear A	Steel	15,215.5	119.4	49.4	81	3.4	Gripper-I	6
10	Gear B	Steel	15,215.5	119.4	49.4	45	3.4	Gripper-I	6
11	Driven Shaft A	Steel	5,207.0	40.9	49.4	81	−7.6	Gripper-I	4
12	Base	Steel	195,539.3	1535.0	49.4	81	49.4	Gripper-II	8
13	Driven Shaft B	Steel	18,267.2	143.4	49.4	45	152.4	Gripper-I	4
14	Packing Gland	Steel	2,709.0	21.3	49.4	45	91.4	Gripper-I	2
15	Gland Nut	Steel	12,046.9	94.6	49.4	45	96.4	Spanner-II	3

Table 3
Gear pump B. Properties and disassembly requirements for all components.

Item	Name	Material	Volume (mm^3)	Weight (g.)	Disassembly point X	Y	Z	Disassembly tool	$t_b(x_i)$ (s)
1	Bolt A	Steel	1,243.1	9.8	59.1	114	−48.4	Spanner-I	4
2	Bolt B	Steel	1,243.1	9.8	90.3	89	−48.4	Spanner-I	4
3	Bolt C	Steel	1,243.1	9.8	90.3	33	−48.4	Spanner-I	4
4	Bolt D	Steel	1,243.1	9.8	59.1	8	−48.4	Spanner-I	4
5	Bolt E	Steel	1,243.1	9.8	27.9	33	−48.4	Spanner-I	4
6	Bolt F	Steel	1,243.1	9.8	27.9	89	−48.4	Spanner-I	4
7	Cover	Steel	95,973.5	753.4	59.1	82	−64.6	Gripper-II	5
8	Gasket	Rubber	5,496.3	5.2	59.1	114	−31.4	Gripper-I	4
9	Gear A	Steel	21,301.7	167.2	59.1	82	−30.9	Gripper-I	6
10	Gear B	Steel	21,301.7	167.2	59.1	40	−30.9	Gripper-I	6
11	Shaft A	Steel	6,430.7	50.5	59.1	40	−48.9	Gripper-I	4
12	Base	Steel	273,755.0	2149.0	59.1	114	7.1	Gripper-II	4
13	Shaft B	Steel	22,560.0	177.1	59.1	82	136.1	Gripper-I	8
14	Gland A	PTFE	3,243.6	7.1	59.1	94.8	34.1	Gripper-I	3
15	Gland B	PTFE	3,243.6	7.1	59.1	94.8	41.1	Gripper-I	3
16	Gland C	PTFE	3,243.6	7.1	59.1	94.8	48.1	Gripper-I	3
17	Gland D	PTFE	3,243.6	7.1	59.1	94.8	55.1	Gripper-I	3
18	Gland E	Steel	14,456.3	113.5	59.1	82	79.1	Gripper-I	3
19	Bolt stud A	Steel	998.1	7.8	35.1	82	89.1	Spanner-II	3
20	Bolt stud B	Steel	998.1	7.8	83.1	82	89.1	Spanner-II	3
21	Nut A	Steel	289.5	2.3	35.1	82	84.1	Spanner-III	4
22	Nut B	Steel	289.5	2.3	83.1	82	84.1	Spanner-III	4
23	Nut C	Steel	289.5	2.3	35.1	82	87.1	Spanner-III	4
24	Nut D	Steel	289.5	2.3	83.1	82	87.1	Spanner-III	4

is similar to an SO approach, as previously explained in Section 3.3. The second consists of a dominance-based MO optimisation approach to generate a set of POSs using MOBA. To compare the solutions, two additional optimisation algorithms, NSGA-II and PESA-II, are used.

The last step of this stage is devoted to the performance analysis. To evaluate the performance of the three MO algorithms, the number of non-dominated solutions generated, HI and NFE, is applied. The number of non-dominated solutions was proposed by Durillo et al. (2010) to measure convergence speed. These criteria cannot be the sole indicator of measurement. A higher number of non-dominated solutions is preferable, but diverse solutions can be achieved through a lower number of non-dominated solutions. Consequently, another indicator must be used to measure the performance of the algorithms. In this way, HI is applied to measure the convergence and diversity of the sets of solutions from the optimal Pareto front (Cao et al., 2015; Talbi, 2009). Linear normalisation is needed to give a relatively equal contribution of objectives to the indicator value (Knowles et al., 2006), as shown in Eq. (10). In this research, the function is normalised in the range [01], and the reference point used is [1.2, 1.2, 1.2].

$$f_{norm} = \frac{f - f_{min}}{f_{max} - f_{min}} \tag{10}$$

In addition, HI is calculated by generating 10,000 samples using Monte Carlo simulation. The higher value of HI is the most preferable, due to its producing a wider range of POSs (Halim et al., 2020). Finally, NFE is applied to measure the speed in finding solutions.

The pseudocode depicting the novel design of the MOBA for the RDSP is displayed in Fig. 5. The first step is to initialise the MOBA parameter setting, Pareto front set, and stopping criteria (maximum number of iterations). The generation of n scoutbees is similar to EDBA, using MFSG to help the generation of feasible disassembly sequences. The n scoutbees are then sorted using Pareto sorting and crowding distance calculation. The process is similar to EDBA for elite sites e and selected sites m. The difference is only that, in each site, the Pareto sorting and crowding distance calculation is applied to sort all the bees. After a random search is performed by the $n - m$ bees to explore the solution space. The Pareto front set is then updated, and the best RDSP information is saved until the stopping criteria are met (maximum iteration reached).

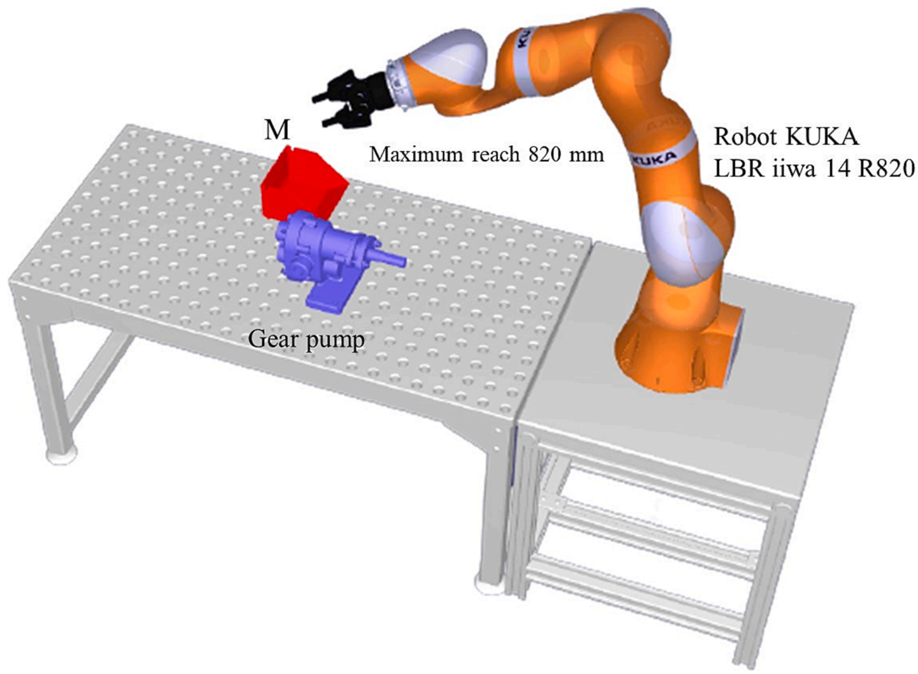


Fig. 8. Layout of the robotic cell.

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.^{a,b}	Decision
1	The distribution of GearA_goal123 is the same across categories of Iter_pop.	Independent-Samples Kruskal-Wallis Test	<.001	Reject the null hypothesis.
2	The distribution of GearB_goal123 is the same across categories of Iter_pop.	Independent-Samples Kruskal-Wallis Test	<.001	Reject the null hypothesis.

a. The significance level is .050.

b. Asymptotic significance is displayed.

Fig. 9. Kruskal-Wallis test results for gear pumps A and B, multiobjective aggregate method.

4. Case studies and experiments

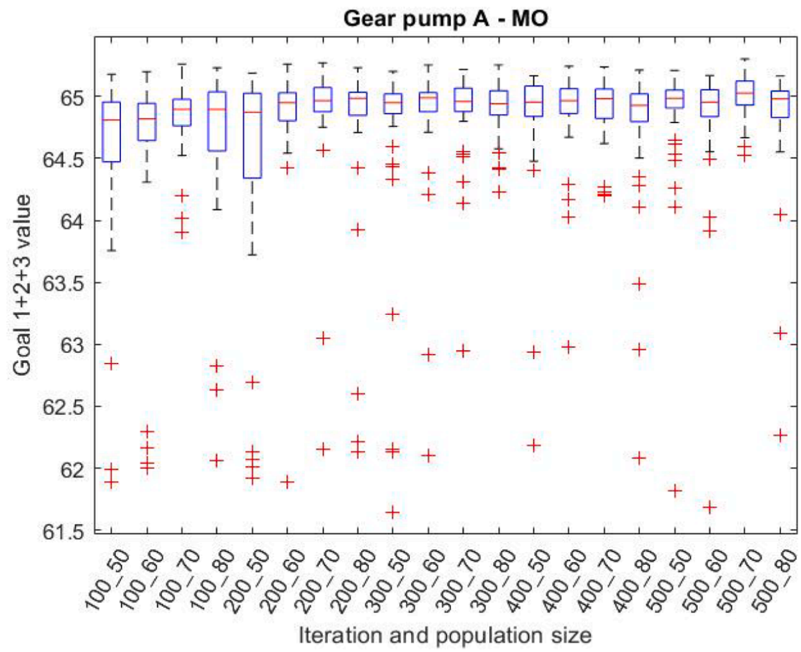
4.1. Case studies

A gear pump is a type of hydraulic pump consisting of two gears enclosed in a tight housing. It transforms kinetic energy in the form of motor torque, generated by a motor, into hydraulic energy through the oil flow generated by the pump. This flow of oil under pressure is normally used to generate the movement of the actuator installed in the machine or application. The main element of the pump is the coupled gear pair. The gear pair is made up of the drive shaft (driven by the motor shaft) and the driven shaft. The driving shaft rotates the driven shaft under the principle of displacement caused by the contact between the teeth of the shaft gears. When the pump is operated, the oil enters through the inlet (suction) hole of the pump due to the depression created by separating the teeth of one gear from those of the other. The oil is transported through the flanks of the gear teeth until it reaches the outlet hole of the pump, where when the teeth of the driving shaft meet those of the driven shaft, the oil is driven towards the outlet hole (pressure). The use of external gear pumps in industry is common because it is a compact, powerful, robust and cost-competitive product. From the perspective of EoL product recovery, the gear pump is of great interest for remanufacturers due to some of its components presenting low wear after its operative lifetime and use, allowing them to be reused or remanufactured to be included in new products. The other components can be recycled or disposed of as the ultimate option.

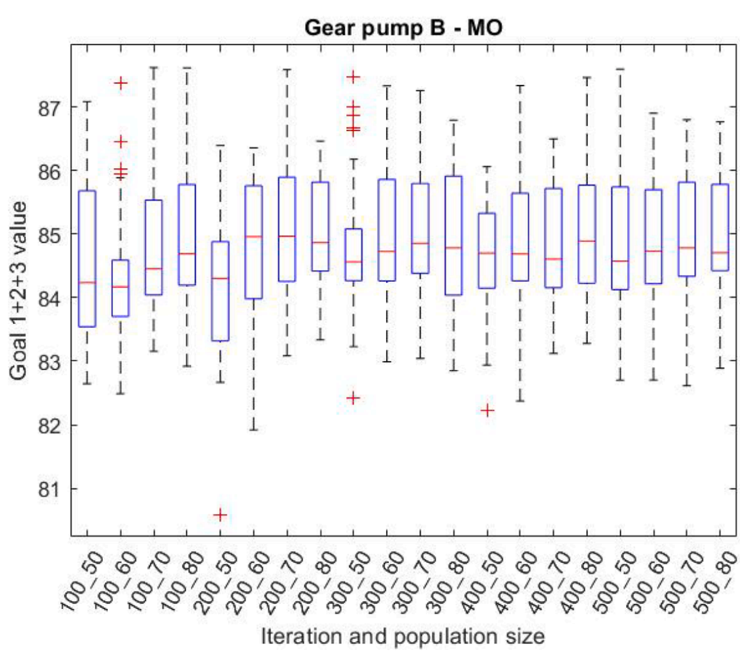
Two external gear pumps were selected as the case studies for this research: Gear Pump A (7.5 l/min) and Gear Pump B (10 l/min), as shown in Figs. 6 and 7. Table 2 and Table 3 show the definition and properties of all the components of both gear pumps. The information was obtained from the 3D models (Grabcad Community, 2020), the works by Liu et al. (2018a) and Ramírez et al. (2020), and specialised remanufacturers in the UK. Additionally, supplementary data (https://doi.org/10.25500/edata.bham.00000810) shows the path distance (PD) matrices for both gear pumps (Tables 1 and 2), for Gear Pump A and B, respectively, indicating the distance between adjacent disassembly points considering all the paths allowed and the infeasible trajectories that prevent forbidden ways from being selected in the disassembly process.

4.2. Robotic cell

A robotic cell made up of a robot and tool magazine (M) with a robotic tool changer was designed for the simulations. Fig. 8 shows a general layout of the robotic cell, designed with the location of the robot, the tool magazine (M), and the selected gear, pump A or B,



(a) MO aggregation method for gear pump A



(b) MO aggregation method for gear pump B

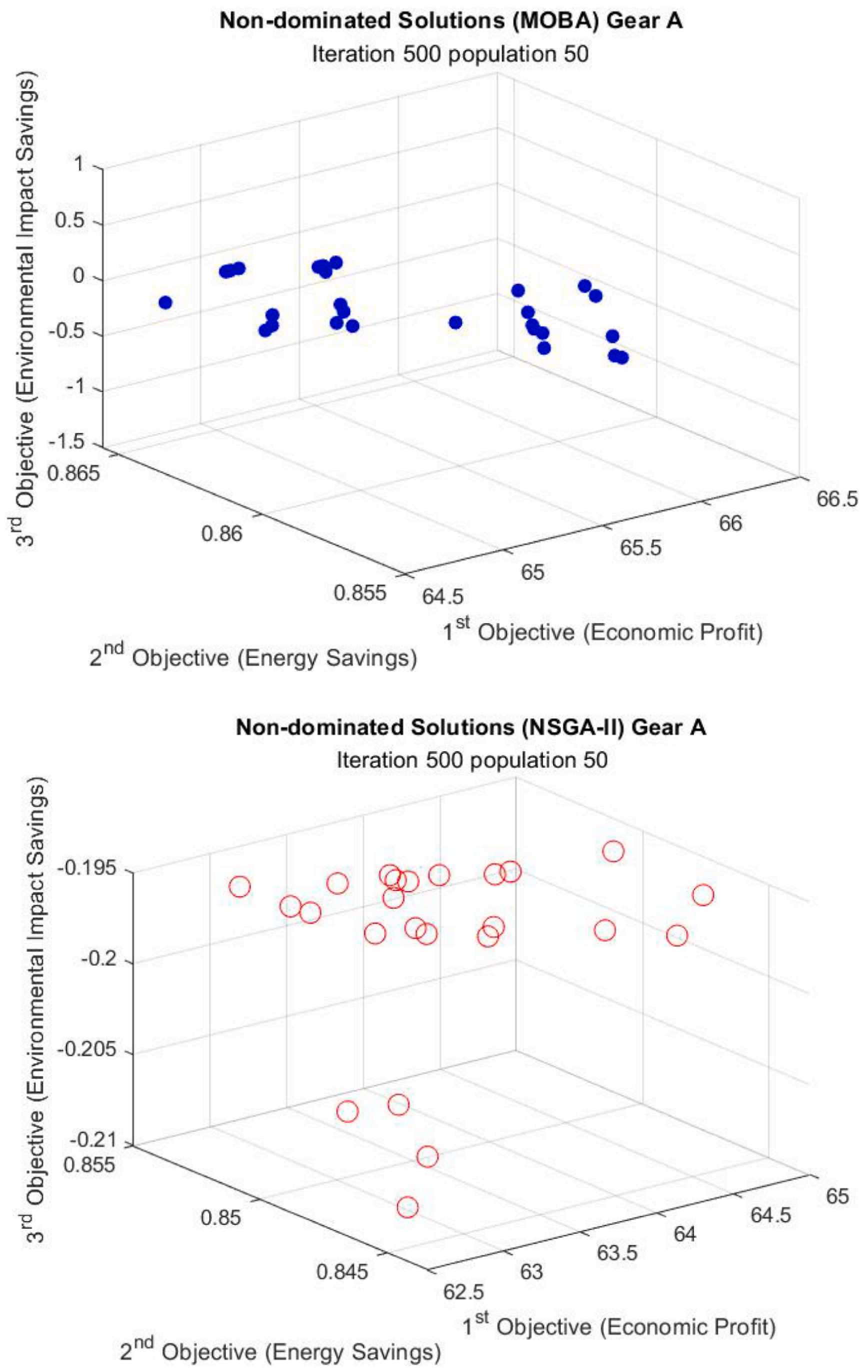
Fig. 10. Boxplot for the multiobjective aggregation method.

according to the case study analysed.

The robot selected was the KUKA LBR iiwa 14 R820 (KUKA, 2020), a 7-axis lightweight robot with a jointed arm. This robot is prepared to manage a rated pay load of up to 14 kg, with a maximum reach of 820 mm. The volume of the working envelope is 1.8 m^3 and the pose repeatability (ISO 9283) is $\pm$ 0.15 mm.

The tool magazine (M) is a device containing the tools needed to perform the disassembly operations. The robot is required to move up to the position of the tool magazine if the subsequent planned step in the disassembly process requires changing the tool. In the simulations, the tool magazine (M) is assumed to be located at the following position: $x = 300$, $y = 200$, $z = 150$.

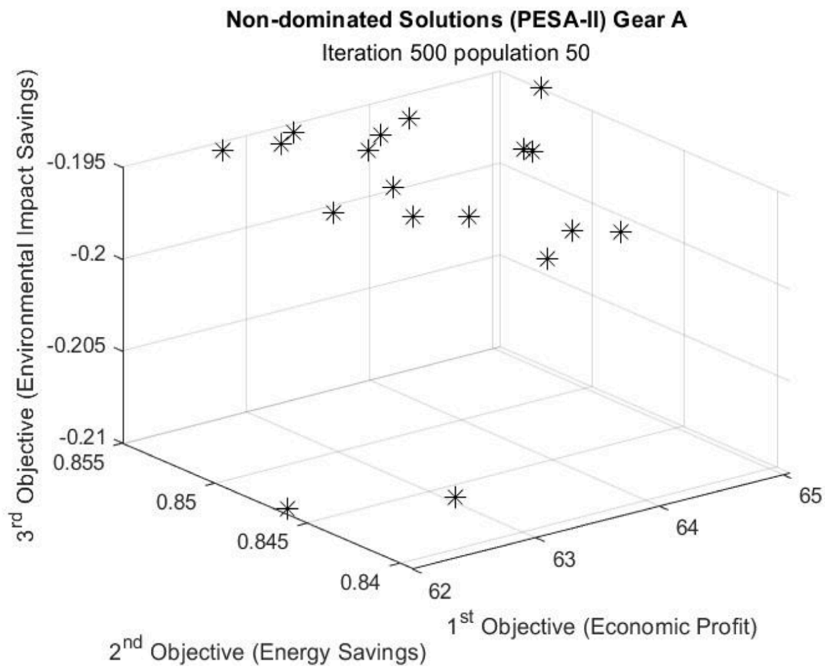
(a) POSs from MOBA



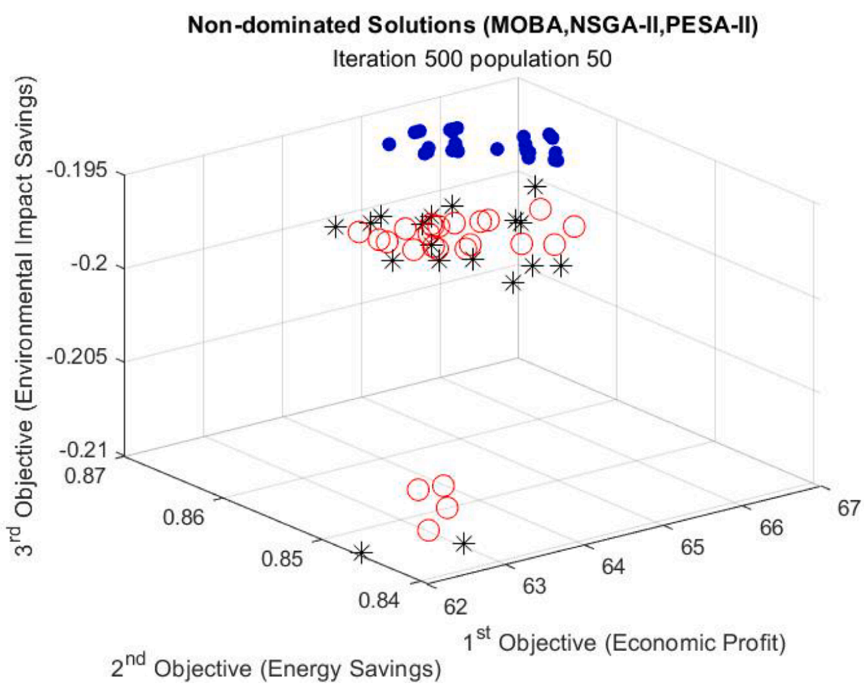
(b) POSs from NSGA-II

Fig. 11. POSs for gear pump A using MOBA, NSGA-II and PESA-II (Iteration 500, population size 50).

The gear pump disassembly process involves two main groups of disassembly operations to be performed: unfastening and pulling/pushing. To unfasten bolts and nuts, three types of spanners are considered (Spanner-I, -II and -III). For the remainder of the disassembly operations, two types of grippers (Gripper-I and -II) are used. Table 2 and Table 3 show the disassembly tool used in each disassembly operation, the coordinates of the disassembly point of each component with relation to the coordinates of the origin, and the basic time, $t_b(x_i)$, needed to complete the disassembly operation.



(c) POSs from PESA-II



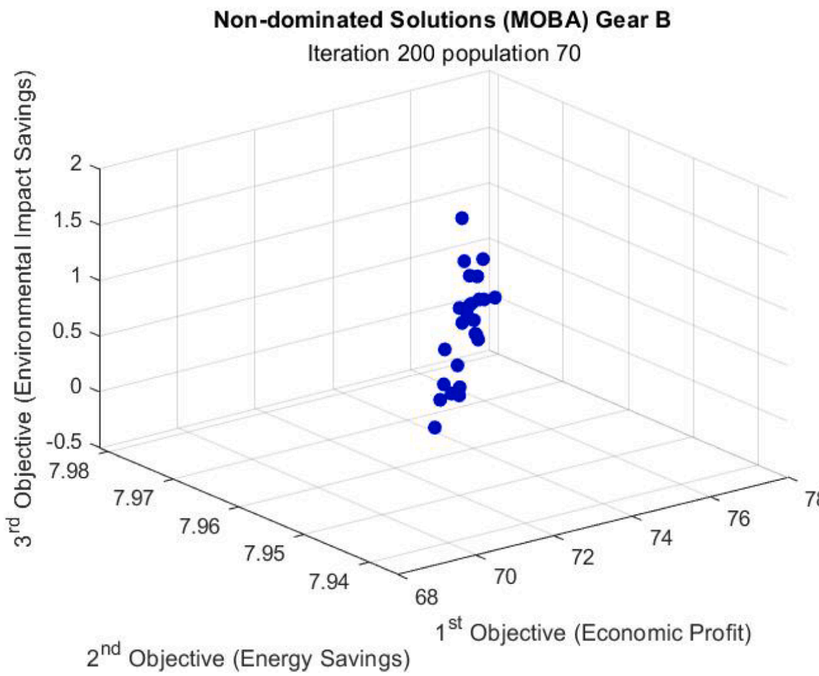
(d) POSs from MOBA, NSGA-II and PESA-II

Fig. 11. (*continued*).

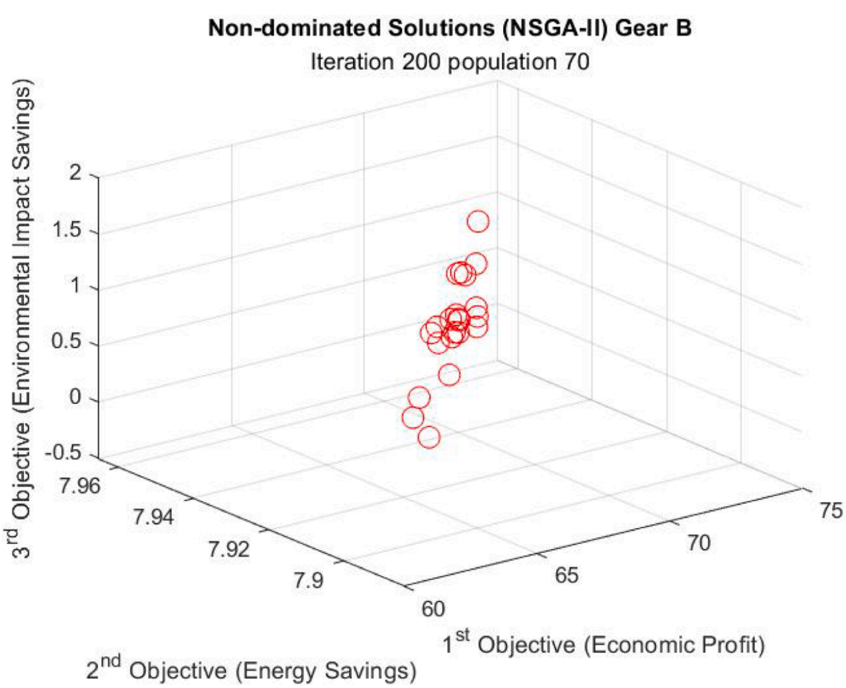
4.3. Key input data and calculation assumptions

To perform the required calculations and simulations, a variety of input data were considered. In supplementary data (https://doi.org/10.25500/edata.bham.00000810), Tables S.3 and S.4 show the input data needed to evaluate the profit goal (f_1) for gear pumps A and B; Tables S.5 to S.8 show the parameters required to assess the energy savings goal (f_2) for both gear pumps, while Tables S.9 to S.12 present the corresponding parameters for the evaluation of the environmental benefits goal (f_3). The authors note that most of the information and data specified in the referred tables were provided by remanufacturing and recycling companies in the UK and Spain during 2020.

In addition to the input data included in the supplementary data (https://doi.org/10.25500/edata.bham.00000810) some calculation

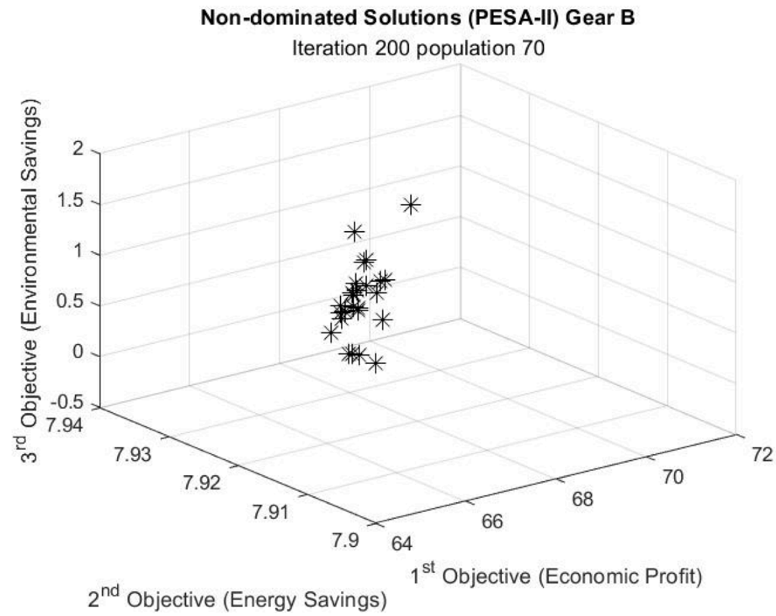


(a) POSs from MOBA

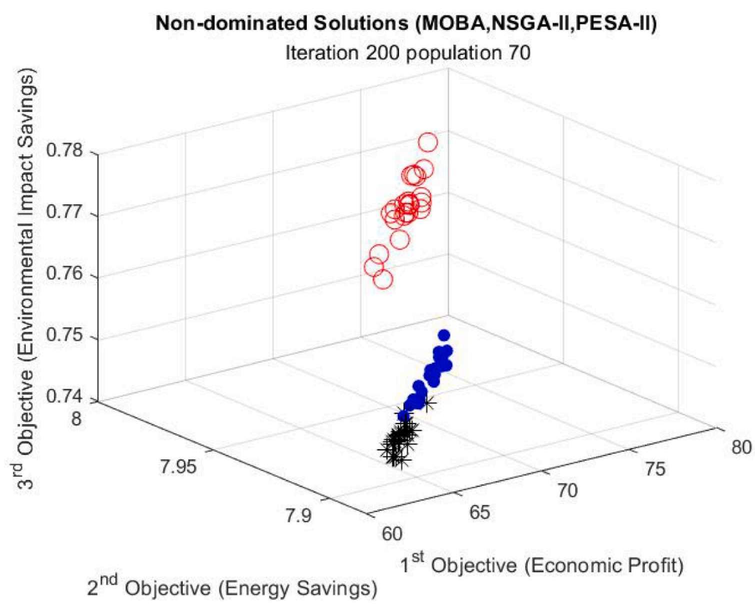


(b) POSs from NSGA-II

Fig. 12. POSs for gear pump B using MOBA, NSGA-II and PESA-II (Iteration 200, population size 70).



(c) POSs from PESA-II



(d) POSs from MOBA, NSGA-II and PESA-II

Fig. 12. (*continued*).

assumptions were considered, as follows:

- The components are disassembled using non-destructive operations. In this way, the research considers the robot is able to complete all the disassembly tasks using standard operations as turning, unscrewing, removal, gripping, etc.
- The task times are deterministic and known. Therefore, same operation is completed by the robot in the same time for all the parts being disassembled.
- The disassembly process is sequential, being the disassembly operations performed one by one.
- The disassembly mode is complete, involving dismantling the whole product into individual components.
- The remanufacturing company is assumed to operate during one 8-hour shift per day, 220 days per year.
- A forecast of the remanufactured gear pumps in the robotic cell is considered equal to 70,000 units per year for Gear Pump A and 55,000 units per year for Gear Pump B, given that the robotic cell is operating with only one type of gear pump all the year. It is based on the information provided by remanufacturers of gear pumps in United Kingdom.

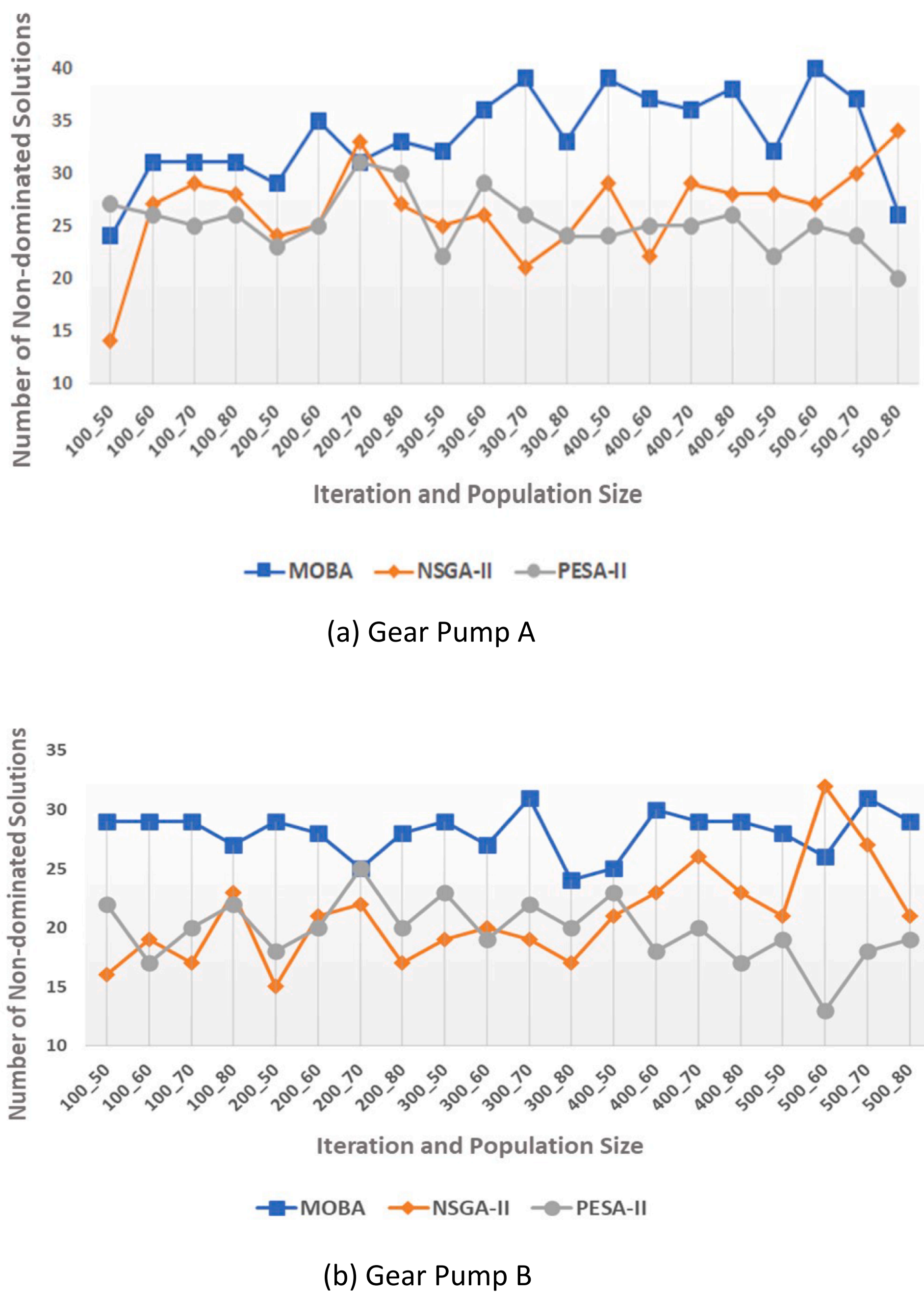


Fig. 13. Number of POSs.

- Based on a commercial quotation of a robot manufacturer, the upfront costs (investment) of the robotic cell are considered to be 0.15 M€, and the hourly cost of the robotic cell is 120 €/h.
- The study considers a straight-line depreciation of the machinery for an expected lifetime of 10 years.
- Overhead costs are distributed according to the resources applied to each disassembly operation. In this way, and following Ramírez et al. (2020), a weight of 2 out of 10 is applied to the components to be reused, 5 out of 10 to the components to be remanufactured, 2 out of 10 to the components to be recycled, and 1 out of 10 to the components to be disposed of.
- Based on the information provided by the manufacturer (KUKA, 2020):
  - o The linear velocity in the movement of the robot's end-effector is equal to 12 mm/s.
  - o The penalty times for process direction changes, p_1 and p_2, are assumed to be equal to 1 and 2 seconds, respectively.
  - o The robot is expected to take 10 seconds to change the tool in the tool magazine (M).

5. Results and discussion

The algorithms presented in Section 3 were programmed and run using MATLAB 2020b on a computer with a 1.80 GHz Intel Core i7-8565U CPU. The statistical tests were conducted using MATLAB 2020b and IBM SPSS Statistics 27. Both the MO aggregate and MO non-dominated approaches were performed using different iteration sizes

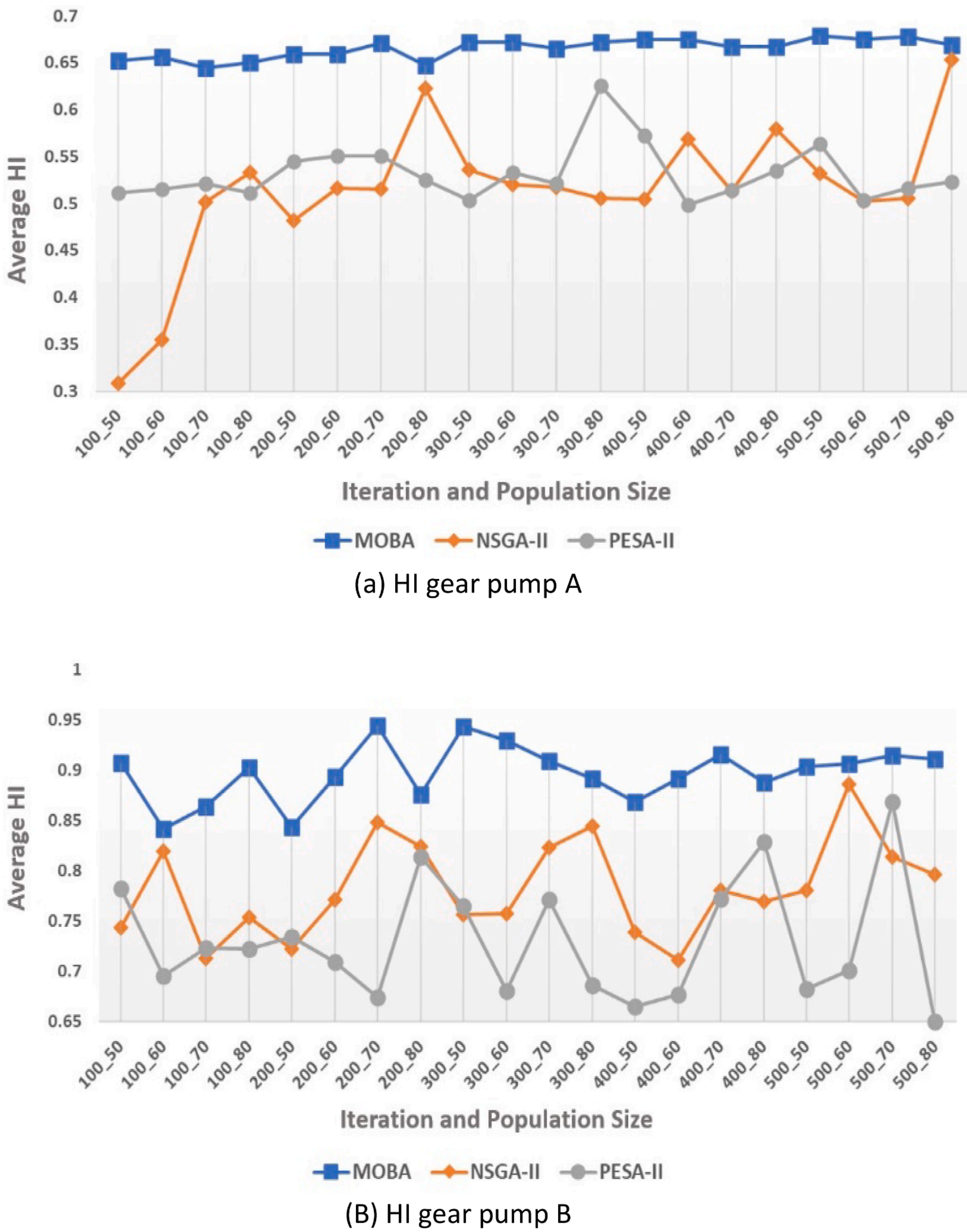


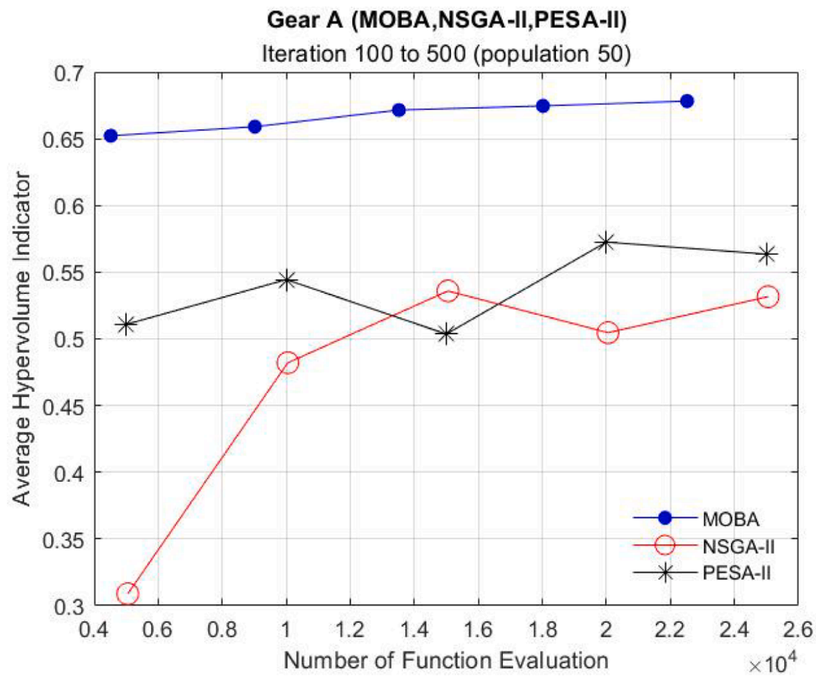
Fig. 14. Hypervolume Indicator using MOBA, NSGA-II and PESA-II.

(100 to 500) and population sizes (50 to 80) and independently run fifty times. The algorithm's stopping criterion is the number of iterations. The Bees Algorithm parameters used in this research were set as follows: number of elite sites (*e*) equal to 1, number of selected sites (*m*) equal to 5, number of recruited bees around elite sites (*nep*) equal to 10, and number of recruited bees around selected sites (*nsp*) equal to 5.

The first step to solve the MO problem in this study consisted of a scalar approach using an aggregation method, where the three goals were added together in a linear manner. The boxplot shown in Fig. 10 presents the results for Gear Pump A, where the maximum fitness value is equal to 65.30 € given by iteration 500 and population size 70.

Likewise, the maximum fitness value for Gear Pump B is 87.62 €, reached by iteration 100 and population size 70. The assumption checklist shows that these data should be processed using the nonparametric Kruskal–Wallis test. The statistical test also rejected the null hypothesis for both Gear Pumps A and B (see Fig. 9), and therefore, the Dunn-Sidak test was carried out. The results show that for Gear Pump A, the difference means are given by iteration 100 with population sizes of 50, 60, 70 and 80 and iteration 200 with population size 50. For Gear Pump B, the difference means are given by iteration 100 with population size 70 and iteration 200 with population size 50. The RDSP results for both gear pumps and the statistical test results are included as supplementary data (https://doi.org/10.25500/edata.bham.00000810) in Tables S.13 and S.14, and Figs. S.1 to S.3.

Dominance-based MO was the second approach used in this study to generate a set of Pareto Optimal Solutions (POSs). The version of the MOBA used in this approach is depicted in Fig. 5 (Section 3). Two additional algorithms, NSGA-II and PESA-II were applied to conduct a comparison between the results. Fig. 11 for Gear Pump A and Fig. 12 for Gear Pump B present the POSs of the trade-off decision between the



(a) Gear pump A - population 50

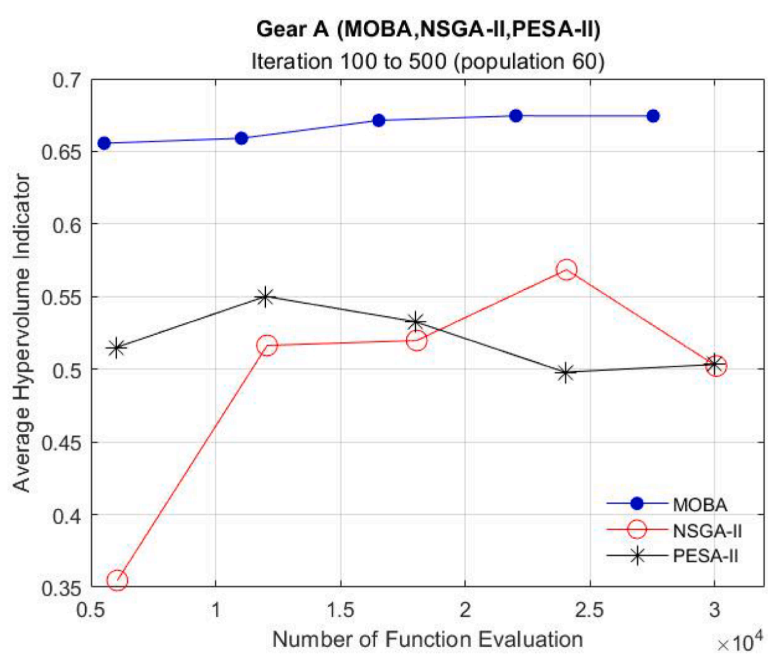
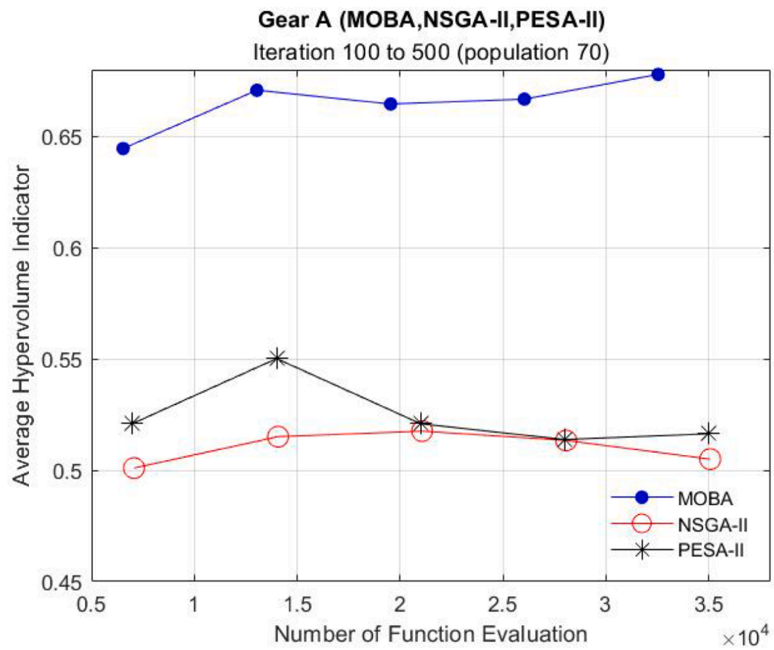


Fig. 15. Number of Function Evaluation for gear pump A, Iteration 100 to 500, population 50, 60, 70 and 80.

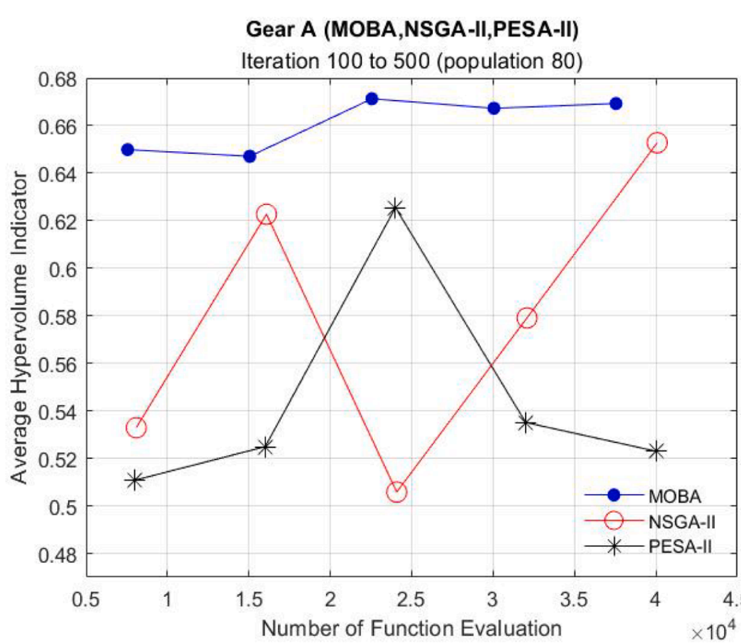
three goals using the three proposed algorithms. Fig. 11d and Fig. 12d depict the aggregation of the results for the three algorithms. The results reveal that NSGA-II and PESA-II yield similar results to MOBA for Gear Pump A, whereas for Gear Pump B, the results are different for the three algorithms. Furthermore, from the analysis of these figures, it is evident that a simple visual comparison is not sufficient to prove that MOBA performs better than the other algorithms. Hence, a performance evaluation to obtain the total number of POSs, HI and NFE was carried out.

Concerning the POSs generated, the line chart shown in Fig. 13 reveals that MOBA yields a higher number of POSs compared to NSGA-II

(b) Gear pump A - population 60



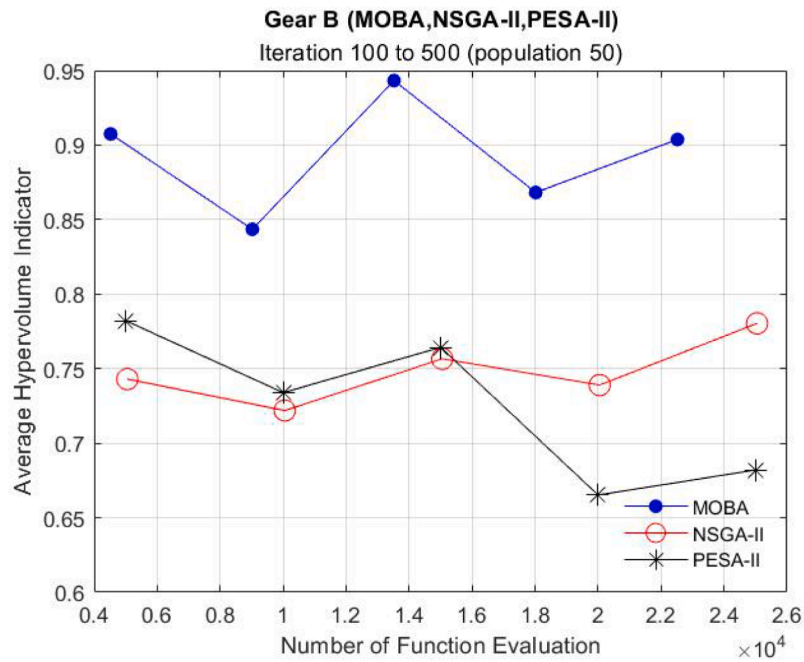
(c) Gear pump A - population 70



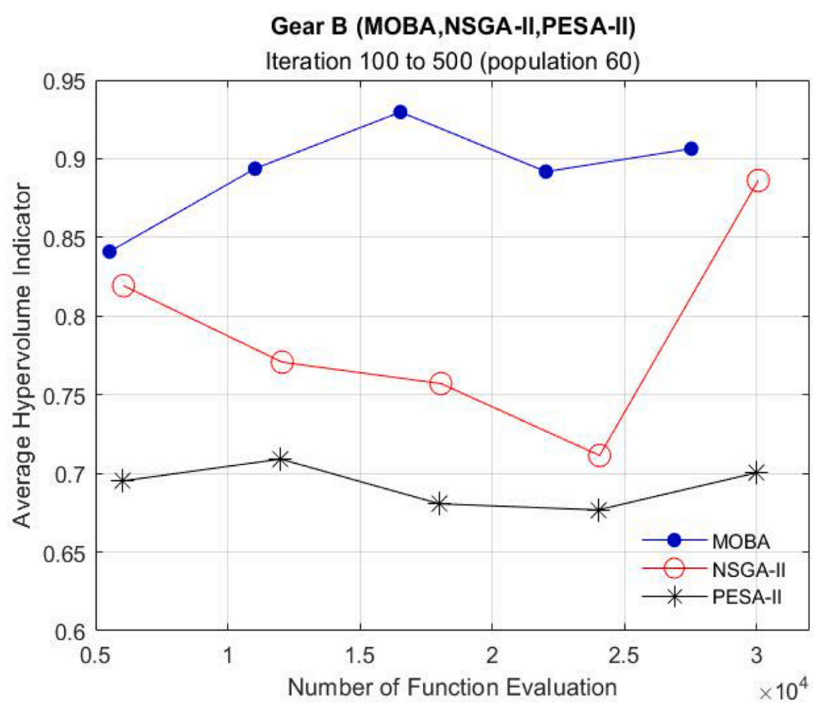
(d) Gear pump A - population 80

Fig. 15. (*continued*).

and PESA-II in most cases and for both gear pumps. MOBA gives a higher number of POSs in 17 iterations for Gear Pump A and 19 iterations for Gear Pump B, from a total of 20 iterations and population sizes used for each case. The higher number of POSs shows the convergence speed, as proposed by Durillo et al. (2010). Moreover, a higher number of POSs means that the decision-maker has more choices and can then easily and



(a) Gear pump B - population 50

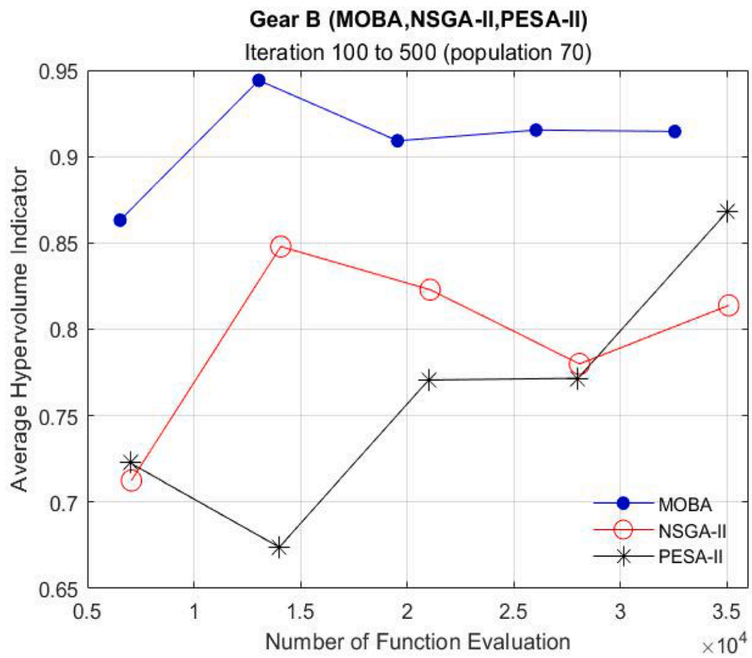


(b) Gear pump B - population 60

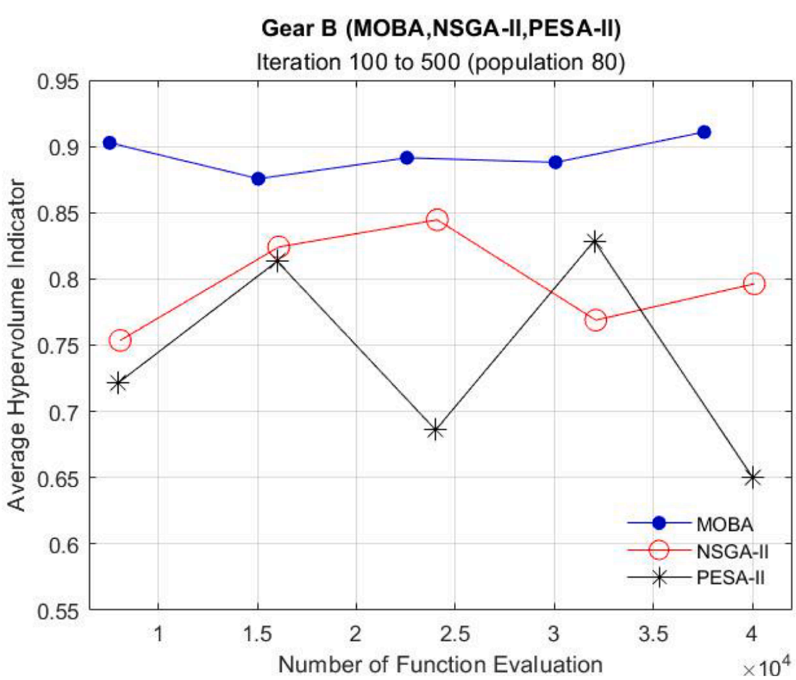
Fig. 16. Number of Function Evaluation for gear pump B, Iteration 100 to 500, population 50, 60, 70 and 80.

accurately adjust the most suitable decisions. Therefore, a higher number of POSs is desirable.

HI is the second performance indicator used in this work. The average HI is compared in each iteration (100 to 500) and population size (50 to 80) for all the experiments. MOBA provides a higher HI than NSGA-II and PESA-II for both gear pumps in every iteration and population size. The HI line graph for gear pump A, depicted in Fig. 14a, shows that MOBA has a steady line compared to NSGA-II and PESA-II. With regard to gear pump B, as shown in Fig. 14b, the HI results reveal that MOBA fluctuates and then levels off after iteration 300 with



(c) Gear pump B - population 70



(d) Gear pump B - population 80

Fig. 16. (*continued*).

population size 70, while NSGA-II and PESA-II present more variations. In conclusion, the MOBA performs better with the calculation of HI (relatively steady), also producing higher HI values.

Likewise, NFE was evaluated for both gear pumps, as shown in Fig. 15 and Fig. 16, using MOBA, NSGA-II and PESA-II. The figures show the relationships between NFE and HI obtained from different iterations with the same population. For both gear pumps, the HI obtained from MOBA gives a steady line, whereas the NSGA-II and PESA-II present greater variation in each graph. Not only does MOBA show a steady behaviour and higher HI in each graph, but it also provides smaller NFEs for all scenarios. The results reveal that MOBA performs better than the other two algorithms due to its ability to yield a higher HI using a smaller NFE for any iteration and population size.

Furthermore, Table 4 and Table 5 show the main RDSP outcomes for

Table 4
Pareto Optimal Solutions of gear pump A (MOBA - Iteration 500, population size 50).

No.	Disassembly Sequence	Disassembly Direction	Disassembly mode	Disassembly Tool	f_1	f_2	f_3
1	1-2-3-6-5-4-15-7-11-9-10-14-13-8-12	2-2-2-2-2-1-2-2-2-2-1-2-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-4-1	1-1-1-1-1-1-2-4-3-3-3-3-3-3-4	65.669	0.8583	-0.195
2	1-6-4-5-3-2-15-7-11-9-10-8-12-14-13	2-2-2-2-2-1-2-2-2-2-2-2-1-1	1-1-1-1-1-1-1-1-1-1-1-4-1-1-1	1-1-1-1-1-1-2-4-3-3-3-3-4-3-3	64.945	0.8627	-0.195
3	1-6-5-4-3-2-7-11-9-10-8-15-12-14-13	2-2-2-2-2-2-2-2-2-2-2-1-2-1-2	1-1-1-1-1-1-1-1-1-1-4-1-1-1-1	1-1-1-1-1-1-4-3-3-3-3-2-4-3-3	65.443	0.8643	-0.195
4	1-6-4-3-2-5-15-7-9-11-10-14-13-8-12	2-2-2-2-2-1-2-2-2-2-1-1-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-4-1	1-1-1-1-1-1-2-4-3-3-3-3-3-3-4	65.696	0.8588	-0.195
5	2-1-3-4-5-6-15-7-9-11-10-8-12-13-14	2-2-2-2-2-1-2-2-2-2-2-2-2-2	1-1-1-1-1-1-1-1-1-1-1-4-1-1-1	1-1-1-1-1-1-2-4-3-3-3-3-4-3-3	64.699	0.8647	-0.195
6	2-1-6-5-4-3-7-10-11-9-8-15-12-13-14	2-2-2-2-2-2-2-2-2-2-2-1-2-2-2	1-1-1-1-1-1-1-1-1-1-4-1-1-1-1	1-1-1-1-1-1-4-3-3-3-3-2-4-3-3	65.192	0.8656	-0.195
7	2-1-6-5-4-3-15-7-10-11-9-13-14-8-12	2-2-2-2-2-1-2-2-2-2-2-1-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-4-1	1-1-1-1-1-1-2-4-3-3-3-3-3-3-4	65.781	0.8595	-0.195
8	2-1-6-5-4-3-7-11-9-10-8-15-12-14-13	2-2-2-2-2-2-2-2-2-2-2-1-2-1-2	1-1-1-1-1-1-1-1-1-1-4-1-1-1-1	1-1-1-1-1-1-4-3-3-3-3-2-4-3-3	65.48	0.8646	-0.195
9	2-4-1-6-3-5-7-9-11-10-8-15-12-13-14	2-2-2-2-2-2-2-2-2-2-2-1-2-2-2	1-1-1-1-1-1-1-1-1-1-4-1-1-1-1	1-1-1-1-1-1-4-3-3-3-3-2-4-3-3	64.805	0.862	-0.195
10	3-2-5-4-6-1-15-7-10-9-11-8-12-14-13	2-2-2-2-2-1-2-2-2-2-2-2-1-1	1-1-1-1-1-1-1-1-1-1-1-4-1-1-1	1-1-1-1-1-1-2-4-3-3-3-3-4-3-3	64.864	0.8622	-0.195
11	3-2-1-5-4-6-7-15-14-13-9-11-10-8-12	2-2-2-2-2-2-1-1-1-2-2-2-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-4-1	1-1-1-1-1-1-4-2-3-3-3-3-3-3-4	65.503	0.8602	-0.195
12	3-4-5-6-1-2-7-10-11-9-8-15-12-14-13	2-2-2-2-2-2-2-2-2-2-2-1-2-1-2	1-1-1-1-1-1-1-1-1-1-4-1-1-1-1	1-1-1-1-1-1-4-3-3-3-3-2-4-3-3	65.456	0.8646	-0.195
13	3-4-5-6-1-2-7-11-9-10-8-15-12-14-13	2-2-2-2-2-2-2-2-2-2-2-1-2-1-1	1-1-1-1-1-1-1-1-1-1-4-1-1-1-1	1-1-1-1-1-1-4-3-3-3-3-2-4-3-3	65.546	0.8646	-0.195
14	3-4-6-1-2-5-15-7-10-11-9-14-13-8-12	2-2-2-2-2-1-2-2-2-2-1-2-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-4-1	1-1-1-1-1-1-2-4-3-3-3-3-3-3-4	65.841	0.8601	-0.195
15	5-1-2-3-4-6-7-11-10-9-8-15-12-14-13	2-2-2-2-2-2-2-2-2-2-2-1-2-1-2	1-1-1-1-1-1-1-1-1-1-4-1-1-1-1	1-1-1-1-1-1-4-3-3-3-3-2-4-3-3	65.242	0.8624	-0.195
16	5-2-4-3-1-6-7-11-9-10-8-15-12-14-13	2-2-2-2-2-2-2-2-2-2-2-1-2-1-2	1-1-1-1-1-1-1-1-1-1-4-1-1-1-1	1-1-1-1-1-1-4-3-3-3-3-2-4-3-3	65.198	0.862	-0.195
17	5-4-3-2-1-6-15-7-10-11-9-14-13-8-12	2-2-2-2-2-1-2-2-2-2-1-1-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-4-1	1-1-1-1-1-1-2-4-3-3-3-3-3-3-4	65.915	0.8608	-0.195
18	5-4-3-2-1-6-7-10-11-9-8-15-12-14-13	2-2-2-2-2-2-2-2-2-2-2-1-2-1-2	1-1-1-1-1-1-1-1-1-1-4-1-1-1-1	1-1-1-1-1-1-4-3-3-3-3-2-4-3-3	65.456	0.8646	-0.195
19	5-4-3-2-1-6-7-11-9-10-8-15-12-13-14	2-2-2-2-2-2-2-2-2-2-2-1-2-2-1	1-1-1-1-1-1-1-1-1-1-4-1-1-1-1	1-1-1-1-1-1-4-3-3-3-3-2-4-3-3	65.149	0.8656	-0.195
20	6-1-2-3-4-5-15-7-9-11-10-13-14-8-12	2-2-2-2-2-1-2-2-2-2-2-1-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-4-1	1-1-1-1-1-1-2-4-3-3-3-3-3-3-4	65.675	0.8586	-0.195
21	6-3-5-1-2-4-15-7-10-9-11-14-13-8-12	2-2-2-2-2-1-2-2-2-2-1-2-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-4-1	1-1-1-1-1-1-2-4-3-3-3-3-3-3-4	65.561	0.8575	-0.195
22	6-1-2-5-4-3-7-15-14-13-9-11-10-8-12	2-2-2-2-2-2-1-1-1-2-2-2-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-4-1	1-1-1-1-1-1-4-2-3-3-3-3-3-3-4	65.472	0.8599	-0.195
23	6-1-2-3-4-5-7-10-11-9-8-15-12-13-14	2-2-2-2-2-2-2-2-2-2-2-1-2-2-1	1-1-1-1-1-1-1-1-1-1-4-1-1-1-1	1-1-1-1-1-1-4-3-3-3-3-2-4-3-3	65.126	0.8656	-0.195
24	6-1-2-3-4-5-7-11-9-10-8-15-12-14-13	2-2-2-2-2-2-2-2-2-2-2-1-2-1-1	1-1-1-1-1-1-1-1-1-1-4-1-1-1-1	1-1-1-1-1-1-4-3-3-3-3-2-4-3-3	65.546	0.8646	-0.195
25	6-1-2-3-4-5-7-11-9-10-8-15-12-13-14	2-2-2-2-2-2-2-2-2-2-2-1-2-2-1	1-1-1-1-1-1-1-1-1-1-4-1-1-1-1	1-1-1-1-1-1-4-3-3-3-3-2-4-3-3	65.149	0.8656	-0.195
26	15-6-1-2-3-4-5-7-10-11-9-14-13-8-12	1-2-2-2-2-2-2-2-2-2-2-1-2-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-4-1	2-1-1-1-1-1-1-4-3-3-3-3-3-3-4	66.121	0.8595	-0.195
27	15-1-2-3-4-5-6-7-10-11-9-14-13-8-12	1-2-2-2-2-2-2-2-2-2-2-1-2-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-4-1	2-1-1-1-1-1-1-4-3-3-3-3-3-3-4	66.162	0.8602	-0.195
28	15-1-2-3-6-4-5-7-9-11-10-13-14-8-12	1-2-2-2-2-2-2-2-2-2-2-2-1-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-4-1	2-1-1-1-1-1-1-4-3-3-3-3-3-3-4	65.728	0.8562	-0.195
29	15-6-4-5-2-3-1-7-11-9-10-8-12-14-13	1-2-2-2-2-2-2-2-2-2-2-2-2-1-1	1-1-1-1-1-1-1-1-1-1-1-4-1-1-1	2-1-1-1-1-1-1-4-3-3-3-3-4-3-3	65.117	0.8611	-0.195
30	15-1-5-4-3-2-6-7-11-9-10-13-14-8-12	1-2-2-2-2-2-2-2-2-2-2-2-1-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-4-1	2-1-1-1-1-1-1-4-3-3-3-3-3-3-4	65.869	0.8572	-0.195
31	15-2-5-4-6-1-3-7-11-9-10-8-12-14-13	1-2-2-2-2-2-2-2-2-2-2-2-2-1-2	1-1-1-1-1-1-1-1-1-1-1-4-1-1-1	2-1-1-1-1-1-1-4-3-3-3-3-4-3-3	65.09	0.8615	-0.195
32	15-6-5-4-3-2-1-7-9-10-11-13-14-8-12	1-2-2-2-2-2-2-2-2-2-2-2-1-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-4-1	2-1-1-1-1-1-1-4-3-3-3-3-3-3-4	65.737	0.856	-0.195

Notes:
Disassembly direction: 1 = Y + direction and 2 = Y- direction.
Disassembly mode: 1 = reuse, 2 = remanufacturing, 3 = recycle, 4 = disposal.
Disassembly tool: 1 = Spanner-I, 2 = Spanner-II, 3 = Gripper-I, 4 = Gripper-II.

both gear pumps, obtained from the POSs using MOBA and the maximum HI values found. These tables show the results with all the information for the disassembly sequence, disassembly direction, disassembly mode, disassembly tool and maximum fitness value achieved for each of the three sustainability goals. Similar outcomes from the POSs using NSGA-II and PESA-II (for the highest HI) are presented in Tables S.15 to S.18 in Hartono et al. (https://doi.org/10.25500/edata.bham.00000810) which provides supplementary material for this article.

To highlight the main outcomes and compare the results for the experimental cases under study, Table 6 shows the maximum fitness values for the near-optimal solutions obtained using the proposed MO approaches.

Analysing the results for the MO aggregate method, it can be observed that higher results are obtained if the three goals are aggregated compared to the results obtained by Goal 1, in which only the economic parameters are included. It can be deduced that the energy and environmental goals must be considered in addition to the traditional economic issues, proving that these noneconomic goals are also able to generate profits for firms.

Furthermore, it can be seen that the MO dominance-based method yields different solutions compared to the other approaches. The maximum results obtained with the aggregate method differ from the dominance-based approach results because the aim of this latter method is to balance each goal, so there is a trade-off between goals. Balancing the three goals using the MO dominance-based approach provides a set of solutions for the decision-maker to select the most suitable choices in each case study.

Finally, from the analysis of the results, the following insights are provided:

- The MOBA offers a greater number of POSs than comparison algorithms, resulting in a faster convergence speed.
- The higher HI produced by the MOBA indicates that the convergence and diversity of the set of solutions are greater than those of the comparing algorithms.
- The lower NFE indicates that MOBA is faster at finding solutions.
- MOBA delivers consistent performance across all of the adjusting parameters (population sizes and maximum number of iterations).
- This research demonstrates that, in addition to the objective of optimising the economic outcome of the disassembly process, the non-economic objectives considered in the work, such as energy savings and reduction of environmental impact, also contribute to the sustainability of the disassembly process from an economic point of view. The paper provides a procedure to measure the non-economic results in monetary units, allowing to demonstrate that robotic disassembly of end-of-life products improves the economic performance of the company from an overall point of view.
- The modelling of the robotic disassembly process proposed in this work becomes a useful decision-making tool. This allows the researcher, or the practitioner, to define the decision-making process adjusting the balance of the sustainable goals or modelling competitive behaviours between goals, assigning more relevance to one goal or another depending on the economic, energy and environmental conditions and the available information of the states of nature.

6. Conclusions

This paper presents a multiobjective optimisation decision-making approach to solve the problem of robotic disassembly of end-of-life products with three sustainability goals: profit, energy savings and

Table 5
Pareto Optimal Solutions of gear pump B (MOBA - Iteration 200, population size 70).

No.	Disassembly Sequence	Disassembly Direction	Disassembly mode	Disassembly Tool	f_1	f_2	f_3
1	1-2-3-5-6-24-23-21-22-4-20-19-18-17-7-13-9-10-8-12-14-16-15-11	2-2-2-2-2-1-1-1-1-2-1-1-1-1-2-1-2-2-2-2-2-1-1-1	1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-1-1-1-1-4-1-4-4-4-1	1-1-1-1-1-3-3-3-3-1-2-2-4-4-5-4-4-4-4-5-4-4-4-4	73.107	7.954	0.7415
2	1-2-4-6-3-23-24-22-21-19-20-5-18-17-7-13-11-9-16-8-10-15-14-12	2-2-2-2-2-1-1-1-1-1-1-2-1-1-2-1-2-2-1-2-2-1-1-1	1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-1-1-1-1-4-4-1-4-4-1	1-1-1-1-1-3-3-3-3-2-2-1-4-4-5-4-4-4-4-4-4-4-4-5	74.145	7.9613	0.7415
3	1-2-6-3-4-5-24-22-23-21-20-19-7-18-17-10-16-8-13-15-14-11-9-12	2-2-2-2-2-2-1-1-1-1-1-1-2-1-1-2-1-2-1-1-1-2-2-1	1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-1-4-4-1-4-4-1-1-1	1-1-1-1-1-1-3-3-3-3-2-2-5-4-4-4-4-4-4-4-4-4-4-5	74.831	7.9615	0.7415
4	1-2-6-3-23-21-24-22-5-20-19-4-18-17-7-16-9-10-13-8-12-15-11-14	2-2-2-2-1-1-1-1-2-1-1-2-1-1-2-1-2-2-1-2-2-1-1-2	1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-1-4-1-1-1-4-1-4-1-4	1-1-1-1-3-3-3-3-1-2-2-1-4-4-5-4-4-4-4-4-5-4-4-4	70.865	7.9432	0.7415
5	1-3-5-4-6-23-24-21-22-2-19-7-20-9-10-8-18-13-12-17-11-16-15-14	2-2-2-2-2-1-1-1-1-2-1-2-1-2-2-2-1-1-2-1-2-1-1-1	1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-1-1-1-4-1-4-4-4	1-1-1-1-1-3-3-3-3-1-2-5-2-4-4-4-4-4-5-4-4-4-4-4	70.496	7.9422	0.7415
6	1-3-5-6-4-2-23-21-24-22-19-20-7-8-18-11-13-9-17-16-15-10-14-12	2-2-2-2-2-2-1-1-1-1-1-1-2-2-1-2-1-2-1-1-1-2-1-2	1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-1-1-1-1-4-4-4-1-4-1	1-1-1-1-1-1-3-3-3-3-2-2-5-4-4-4-4-4-4-4-4-4-4-5	76.096	7.9709	0.7415
7	1-6-5-2-24-23-21-22-3-4-20-19-18-17-13-7-16-9-8-10-12-15-11-14	2-2-2-2-1-1-1-1-2-2-1-1-1-1-1-2-1-2-2-2-2-1-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-1-1-4-1-4-1-1-4-1-4	1-1-1-1-3-3-3-3-1-1-2-2-4-4-4-5-4-4-4-4-5-4-4-4	70.549	7.9413	0.7415
8	2-1-6-5-3-23-24-21-22-20-19-4-7-18-11-13-10-8-17-9-16-15-14-12	2-2-2-2-2-1-1-1-1-1-1-2-2-1-2-1-2-2-1-2-1-1-1-1	1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-4-1-4-4-4-1	1-1-1-1-1-3-3-3-3-2-2-1-5-4-4-4-4-4-4-4-4-4-4-5	74.503	7.962	0.7415
9	2-3-5-4-1-24-22-6-20-23-21-19-7-8-10-9-18-13-12-14-11-17-15-16	2-2-2-2-2-1-1-2-1-1-1-1-2-2-2-2-1-1-2-2-1-1-2-1	1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-1-1-1-1-1-4-1-4-4-4	1-1-1-1-1-3-3-1-2-3-3-2-5-4-4-4-4-4-5-4-4-4-4-4	68.936	7.9354	0.7415
10	2-3-1-5-4-23-24-22-21-6-19-20-7-8-9-18-10-11-17-16-15-14-13-12	2-2-2-2-2-1-1-1-1-2-1-1-2-2-2-1-2-2-1-1-1-1-1-2	1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-1-1-1-1-4-4-4-4-1-1	1-1-1-1-1-3-3-3-3-1-2-2-5-4-4-4-4-4-4-4-4-4-4-5	73.147	7.9544	0.7415
11	2-4-3-5-6-23-21-24-22-1-20-19-18-17-13-7-16-11-8-9-15-10-14-12	2-2-2-2-2-1-1-1-1-2-1-1-1-1-1-2-1-2-2-2-1-2-1-2	1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-1-1-4-1-4-1-4-1-4-1	1-1-1-1-1-3-3-3-3-1-2-2-4-4-4-5-4-4-4-4-4-4-4-5	73.822	7.9598	0.7415
12	3-1-2-6-5-4-24-23-21-22-20-19-18-7-8-17-16-9-10-11-15-14-13-12	2-2-2-2-2-2-1-1-1-1-1-1-1-2-2-1-1-2-2-2-1-1-1-2	1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-4-4-1-1-1-4-4-1-1	1-1-1-1-1-1-3-3-3-3-2-2-4-5-4-4-4-4-4-4-4-4-4-5	74.214	7.9614	0.7415
13	3-6-5-1-2-24-22-23-21-20-19-4-7-18-10-8-17-9-13-11-16-15-14-12	2-2-2-2-2-1-1-1-1-1-1-2-2-1-2-2-1-2-1-2-1-1-1-2	1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-4-1-1-1-4-4-4-1	1-1-1-1-1-3-3-3-3-2-2-1-5-4-4-4-4-4-4-4-4-4-4-5	74.581	7.9617	0.7415
14	4-2-5-1-6-3-24-23-21-22-19-20-18-17-16-7-8-15-10-9-13-14-11-12	2-2-2-2-2-2-1-1-1-1-1-1-1-1-1-2-2-1-2-2-1-1-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-4-1-4-4-1-1-1-4-1-1	1-1-1-1-1-1-3-3-3-3-2-2-4-4-4-5-4-4-4-4-4-4-4-5	75.35	7.9673	0.7415
15	4-3-6-5-1-2-23-24-22-21-20-7-19-9-10-18-8-13-12-14-15-11-16-17	2-2-2-2-2-2-1-1-1-1-1-2-1-2-2-1-2-1-2-2-2-2-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-1-1-4-4-1-4-4	1-1-1-1-1-1-3-3-3-3-2-5-2-4-4-4-4-4-5-4-4-4-4-4	71.661	7.9483	0.7415
16	4-5-3-24-23-22-21-19-20-1-6-2-7-18-9-10-13-8-17-16-12-14-11-15	2-2-2-1-1-1-1-1-1-2-2-2-2-1-2-2-1-2-1-1-2-2-1-1	1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-4-4-1-4-1-4	1-1-1-3-3-3-3-2-2-1-1-1-5-4-4-4-4-4-4-4-5-4-4-4	73.343	7.9576	0.7415
17	4-6-5-3-2-1-24-22-23-21-19-20-7-18-8-9-10-11-17-16-15-14-13-12	2-2-2-2-2-2-1-1-1-1-1-1-2-1-2-2-2-2-1-1-1-1-1-1	1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-1-1-1-4-4-4-4-1-1	1-1-1-1-1-1-3-3-3-3-2-2-5-4-4-4-4-4-4-4-4-4-4-5	75.716	7.9715	0.7415
18	5-3-1-6-4-2-24-23-22-21-19-20-7-8-10-9-18-17-11-13-16-15-12-14	2-2-2-2-2-2-1-1-1-1-1-1-2-2-2-2-1-1-2-1-1-1-2-1	1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-1-1-1-4-1-1-4-4-1-4	1-1-1-1-1-1-3-3-3-3-2-2-5-4-4-4-4-4-4-4-4-4-5-4	75.252	7.9679	0.7415
19	5-4-6-2-24-23-21-22-20-19-3-18-17-1-7-13-16-11-9-15-10-8-14-12	2-2-2-2-1-1-1-1-1-1-2-1-1-2-2-1-1-2-2-1-2-2-1-2	1-1-1-1-1-1-1-1-1-1-1-1-1-4-1-1-1-4-1-1-4-1-4-4-1	1-1-1-1-3-3-3-3-2-2-1-4-4-1-5-4-4-4-4-4-4-4-4-5	73.636	7.9576	0.7415
20	5-4-1-3-6-2-24-23-21-22-20-19-18-17-16-7-13-8-10-11-9-15-14-12	2-2-2-2-2-2-1-1-1-1-1-1-1-1-1-2-1-2-2-2-2-1-1-2	1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-4-1-1-4-1-1-1-4-4-1	1-1-1-1-1-1-3-3-3-3-2-2-4-4-4-5-4-4-4-4-4-4-4-5	77.308	7.9814	0.7415
21	5-6-1-3-2-23-21-24-22-4-19-20-7-18-17-16-10-9-13-15-8-12-14-11	2-2-2-2-2-1-1-1-1-2-1-1-2-1-1-1-2-2-1-1-2-2-1-1	1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-4-1-1-1-4-4-1-4-1	1-1-1-1-1-3-3-3-3-1-2-2-5-4-4-4-4-4-4-4-4-5-4-4	72.971	7.953	0.7415
22	6-3-1-4-2-23-21-24-5-19-7-22-20-18-17-8-9-10-13-16-12-14-11-15	2-2-2-2-2-1-1-1-2-1-2-1-1-1-1-2-2-2-1-1-2-2-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-4-1-1-1-4-1-4-1-4	1-1-1-1-1-3-3-3-1-2-5-3-2-4-4-4-4-4-4-4-5-4-4-4	70.074	7.9414	0.7415
23	6-5-4-1-2-23-24-22-21-3-19-7-20-18-8-17-10-9-13-16-12-14-11-15	2-2-2-2-2-1-1-1-1-2-1-2-1-1-2-1-2-2-1-1-2-2-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-4-1-1-1-4-1-4-1-4	1-1-1-1-1-3-3-3-3-1-2-5-2-4-4-4-4-4-4-4-5-4-4-4	72.057	7.9526	0.7415
24	6-5-3-24-22-23-21-19-20-2-1-4-18-7-8-9-13-10-12-17-14-11-15-16	2-2-2-1-1-1-1-1-1-2-2-2-1-2-2-2-1-2-2-1-2-1-2-2	1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-1-1-1-1-4-4-1-4-4	1-1-1-3-3-3-3-2-2-1-1-1-4-5-4-4-4-4-5-4-4-4-4-4	70.718	7.9447	0.7415
25	24-4-3-5-6-1-2-23-21-22-20-19-18-17-16-7-15-11-13-8-14-10-9-12	1-2-2-2-2-2-2-1-1-1-1-1-1-1-1-2-1-2-1-2-1-2-2-1	1-1-1-1-1-1-1-1-1-1-1-1-1-1-4-4-1-4-1-1-4-4-1-1-1	3-1-1-1-1-1-1-3-3-3-2-2-4-4-4-5-4-4-4-4-4-4-4-5	73.864	7.9612	0.7415

Notes:
Disassembly direction: 1 = Y + direction and 2 = Y- direction.
Disassembly mode: 1 = reuse, 2 = remanufacturing, 3 = recycle, 4 = disposal.
Disassembly tool: 1 = Spanner-I, 2 = Spanner-II, 3 = Spanner- III, 4 = Gripper-I, 5 = Gripper-II.

environmental benefits. The proposed model considers all the factors involved in the robotic disassembly of the components and their subsequent recovery option—reuse, remanufacturing, recycling or disposal—in an attempt to maximise the three goals in terms of economic outcomes and reducing in energy and environmental impact.

The model designed is able to find the optimal or near-optimal solutions, obtaining the maximum fitness values for the three sustainability goals and providing a balance between them, using the Pareto front to make the best trade-off decisions. The disassembly sequence, disassembly directions, disassembly tools, and the most suitable recovery option for the disassembled components are the main outcomes of the model. This study introduces a novel approach in the area of RDSP by leveraging the BA autonomously to identify the optimal recovery strategy for each disassembly component while considering constraints and effectively balancing objectives, filling a gap in prior RDSP research. Real case studies based on two models of industrial gear pumps were used as demonstrators. The definition of these case studies was supported by extensive and updated information obtained from the literature and companies specialised in remanufacturing. The experiments conducted for the proposed case studies were simulated in a robotic cell with the appropriate disassembly tools and feasible robot paths and directions, demonstrating the suitability and effectiveness of the model. Therefore, the outcome of the simulated solutions resembles the actual robotic disassembly process.

Furthermore, our research has several implications for both researchers and industrial practitioners. First, this is a pioneering study using two different multiobjective approaches in robotic disassembly. The results show that optimal solutions based on the multiobjective aggregate approach, or the combination and balance of these in a multiobjective methodology, provide greater choices for decision-makers to select the best strategy according to their preferences. Second, the results show that MOBA performs better than NSGA-II and PESA-II in

Table 6
Summary of the best near-optimal solutions.

Gear pump	Bees Algorithm			MOBA	NSGA-II	PESA-II
A	Goal 1 + 2 + 3	65.305 €	Goal 1	66.162 €	65.946 €	65.658 €
			Goal 2	0.866 €	0.862 €	0.861 €
			Goal 3	–0.195 €	−0.195 €	−0.195 €
				HI = 0.6782	HI = 0.6528	HI = 0.6255
B	Goal 1 + 2 + 3	87.625 €	Goal 1	77.308 €	75.187 €	75.523 €
			Goal 2	7.981 €	7.967 €	7.970 €
			Goal 3	0.741 €	0.777 €	0.741 €
				HI = 0.9441	HI = 0.8862	HI = 0.8681

Multiobjective Aggregate Method Multiobjective Dominance-based Approach.

Table A1
Acronyms.

ABC	Artificial Bee Colony
ACO	Ant Colony Optimisation
BA	Bees Algorithm
CAD	Computer Aided Design
CE	Circular Economy
DIS	Disposal
DSP	Disassembly Sequence Planning
EDBA	Enhanced Discrete Bees Algorithm
EDBA-WMO	Enhanced Discrete Bees Algorithm without Mutation Operator
EoL	End-of-life
FA	Flatworm Algorithm
FFO	Fruit Fly Optimisation
GA	Genetic Algorithm
GA-PPX	Genetic Algorithm with Precedence Preserve Crossover
HI	Hypervolume Indicator
IA	Immune Algorithm
LCD	Liquid Crystal Display
MFSG	Modified Feasible Solution Generation
M	Tools magazine
MO	Multiobjective
MOBA	Multiobjective Bees Algorithm
MOEA	Multiobjective Evolutionary Algorithm
MOGA	Multiobjective Genetic Algorithm
MOMVO	Multiobjective Multiverse Optimiser
MRO	Material Recovery Opportunities
NFE	Number of Function Evaluation
NP	Non-Deterministic Polynomial
NSGA-II	Non-dominated Sorting Genetic Algorithm - II
OEM	Original Equipment Manufacturer
OS	Optimal Solution
PD	Path Distance
PESA-II	Pareto Envelope based Selection Algorithm - II
POSs	Pareto Optimal Solutions
PSO	Particle Swarm Optimisation
RDP	Robotic Disassembly Process
RDSP	Robotic Disassembly Sequence Planning
REC	Recycling
REM	Remanufacturing
REU	Reuse
RIC	Remanufacturing Industries Council
SA	Simulated Annealing
SASSO	Self-Adaptive Simplified Swarm Optimisation
SO	Single-objective
TLBO	Teaching Learning Based Optimisation

diversity and convergence speed to find near-optimal results for the RDSP. Third, this research uses statistical methodology to identify the most appropriate parameters to select the combination of numbers of iterations and population sizes that best fit the optimal or near-optimal solutions for the RDSP problem, something which no previous works have done. Fourth, the model allows the disassembly operations to be simulated with great similarity to the real processes. This is of interest for industrial practitioners from the perspective of the required investments and the operations management issues involved in the robotic disassembly process to be implemented in the factory. Fifth, the practical approach of this research is unquestionable, as is the great help this tool could provide for companies in managing EoL products, optimising disassembly processes and achieving sustainability goals.

Future research could extend this study in three ways. First, researchers could study how the model best performs considering uncertainties in the initial condition of the components to be disassembled, taking into account that not all the products present the same state after their useful life. Second, the model could be applied to partial or selective disassembly processes, providing solutions that obtain the optimal disassembly level or stopping point of the process that achieves the best outcomes in terms of sustainability. Third, a mixed integer linear model could be developed in order to obtain optimal solutions for the RDSP problem using mathematical programming solvers.

Table A2
Parameters.

CD_i	Disposal cost of the component i
c_T	Cost per unit of time
$dp_{i,j}$	Depreciation cost
e	Number of elite sites
$er_{i,j}$	Reclaimed environmental benefits from the component i
$ec_{i,j}$	Environmental benefits in the recovering process of component i with mode j
$ed(x_i)$	Environmental benefits in the disassembly operation x_i
$ed(x_i, x_{i+1})$	Environmental benefits between x_i and x_{i+1}
f_1	Profit goal
f_2	Energy savings goal
f_3	Environmental benefits goal
f_W	Conversion factor from kWh to monetary units
$gd_{1,i}(x_i)$	Energy consumption of the robot in the operation x_i
$gd_{2,i}(x_i, M)$	Energy consumption of the robot from x_i to M
$gd_{3,i}(M)$	Energy consumption of the robot in the tool change
$gd_{4,i}(M, x_{i+1})$	Energy consumption of the robot from M to x_{i+1}
$gd_{5,i}(x_i, x_{i+1})$	Energy consumption of the robot from x_i to x_{i+1}
$gc_{i,j}$	Energy consumption for recovering component i with mode j
$gr_{i,j}$	Reclaimed energy
i	Indicator of component
j	Indicator of recovery mode
m	Number of selected sites
n	Number of scout bees
nep	Number of recruited bees around elite sites
nsp	Number of recruited bees around selected sites
N	Total number of components
$oh_{i,j}$	Overhead cost
$PD(x_i, M)$	Path distance from operation x_1 to M
$PD(M, x_{i+1})$	Path distance from M to operation x_{i+1}
$PD(x_i, x_{i+1})$	Path distance from operation x_i to operation x_{i+1}
PR_1	Electric power of the robot used in the disassembly operation
PR_2	Electric power of the robot used in the movements between the disassembly points
$r_{i,j}$	Indicator of recovery mode
$rc_{i,j}$	Recovering cost
RC_i	Recycling revenue of component i
RP_i	Retail price of the component i
$t_b(x_i)$	Basic time of the operation x_i
$t_c(x_i, x_{i+1})$	Tool change time
$t_u(x_i, M)$	Penalty time in the path between operation x_i and M
$t_w(M, x_{i+1})$	Penalty time in the path between M and operation x_{i+1}
$t_z(x_i, x_{i+1})$	Penalty time in the path between operation x_i and operation x_{i+1}
v_e	Line velocity of the industrial robot's end effector
x_i	Disassembly operation of component i
x_{i+1}	Disassembly operation following to x_i
α_i	Indicator equal to 1 if the component i is disassembled, and 0 otherwise
γ_i	Indicator equal to 1 if x_{i+1} requires to change the tool, and 0 otherwise

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Supplementary data: https://doi.org/10.25500/edata.bham.00000810.

Acknowledgements

This work was partially supported by the Engineering and Physical Sciences Research Council (EPSRC), UK, grant no. EP/N018524/1. Natalia Hartono's doctoral study is supported by the Indonesian Endowment Fund for Education (LPDP), Ministry of Finance, Republic of Indonesia, grant no. 201908222215155 under BUDI LN scheme.

Appendix A. Nomenclature and acronyms

Notations and related descriptions for acronyms and parameters are presented in Tables A.1 and A.2.

References

Agrawal, D., & Pande, S. (2011). *Automatic disassembly sequence planning and optimization*. IIT Bombay. Dual Degree Thesis.

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*, 69–79.

Alshibli, M., El Sayed, A., Tozanli, O., Kongar, E., Sobh, T. M., & Gupta, S. M. (2018). A decision maker- centered end-of-life product recovery system for robot task sequencing. *Journal of Intelligent & Robotic Systems, 91*, 603–616.

Azab, A., Ziout, A., ElMaraghy, W., 2011. Modeling and optimization for disassembly planning. JJMIE5.

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.

Brennan, L., Gupta, S. M., & Taleb, K. N. (1994). Operations planning issues in an assembly/disassembly environment. *International Journal of Operations & Production Management*.

Cao, Y., Smucker, B. J., & Robinson, T. J. (2015). On using the hypervolume indicator to compare Pareto fronts: Applications to multi-criteria optimal experimental design. *Journal of Statistical Planning and Inference, 160*, 60–74.

Chen, B., Xu, W., Liu, J., Ji, Z., Zhou, Z., 2020. Robotic disassembly sequence planning considering robotic collision avoidance trajectory in remanufacturing, in: 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), IEEE. pp. 494 – 501.

Chiodo, J., & Ijomah, W. L. (2014). Use of active disassembly technology to improve remanufacturing productivity: Automotive application. *International Journal of Computer Integrated Manufacturing, 27*, 361–371.

Chunming, Z. (2016). Optimization for disassemble sequence planning of electromechanical products during recycling process based on genetic algorithms. *International Journal of Multimedia and Ubiquitous Engineering, 11*, 107–114.

Deb, K., & Jain, H. (2013). An evolutionary many-objective optimization algorithm using reference-point- based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation, 18*, 577–601.

Dong, J., & Arndt, G. (2003). A review of current research on disassembly sequence generation and computer aided design for disassembly. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 217*, 299–312.

Durillo, J., Nebro, A., Luna, F., Coello Coello, C., & Alba, E. (2010). Convergence speed in multi-objective metaheuristics: Efficiency criteria and empirical study. *International Journal for Numerical Methods in Engineering, 84*, 1344–1375.

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, 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*, 43–52.

Fan, S. K. S., Fan, C., Yang, J. H., & Liu, K. F. R. (2013). Disassembly and recycling cost analysis of waste notebook and efficiency improvement by re-design process. *Journal of Cleaner Production, 39*, 209–219.

Fu, Y., Zhou, M., Guo, X., Qi, L., Sedraoui, K., 2021. Multiverse optimization algorithm for stochastic biological disassembly sequence planning subject to operation failures. IEEE Transactions on Systems, Man, and Cybernetics: Systems.

Gao, Y., Wang, Q., Feng, Y., Zheng, H., Zheng, B., & Tan, J. (2018). An energy-saving optimization method of dynamic scheduling for disassembly line. *Energies, 11*, 1261.

Geissdoerfer, M., Savaget, P., Bocken, N. M., & Hultink, E. J. (2017). The circular economy–a new sustainability paradigm? *Journal of Cleaner Production, 143*, 757–768.

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*, 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*, 77–95.

Gonnuru, V. K., et al. (2013). Disassembly planning and sequencing for end-of-life products with RFID-enriched information. *Robotics and Computer-Integrated Manufacturing, 29*, 112–118.

Grabcad Community, 2020. Gear pump 10 l/min. URL: https://grabcad.com/library/gear-pump-10 l-min-1(Accessed: Jul22,2020).

Gunji, B. M., Pabba, S. K., Rajaram, I. R. S., Sorakayala, P. S., Dubey, A., Deepak, B., et al. (2021). Optimal disassembly sequence generation and disposal of parts using stability graph cut-set method for end of life product. *Sadhana, 46*, 1–15.

Halim, A. H., Ismail, I., & Das, S. (2020). Performance assessment of the metaheuristic optimization algorithms: An exhaustive review. *Artificial Intelligence Review*, 1–87.

Han, H. J., Yu, J. M., & Lee, D. H. (2013). Mathematical model and solution algorithms for selective disassembly sequencing with multiple target components and sequence-dependent setups. *International Journal of Production Research, 51*, 4997–5010.

Hartono, N., Ramírez, F.J., Pham, D.T., 2022a. Optimisation of robotic disassembly sequence plans for sustainability using the multi-objective bees algorithm, in: Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. 1 ed.. Springer. Springer Series in Advanced Manufacturing. chapter 19, pp. 337–363.

Hartono, N., Ramírez, F. J., & Pham, D. T. (2022b). Optimisation of robotic disassembly plans using the bees algorithm. *Robotics and Computer-Integrated Manufacturing, 78*, Article 102411.

Hartono, N., Ramírez, F. J., & Pham, D. T. (2022c). A sustainability-based model for robotic disassembly sequence planning in remanufacturing using the bees algorithm. *IFAC-PapersOnLine, 55*, 1013–1018.

Huang, H. H. T., Wang, M. H., & Johnson, M. R. (2000). Disassembly sequence generation using a neural network approach. *Journal of Manufacturing Systems, 19*, 73–82.

Hula, A., Jalali, K., Hamza, K., Skerlos, S. J., & Saitou, K. (2003). Multi-criteria decision-making for optimization of product disassembly under multiple situations. *Environmental Science and Technology, 37*, 5303–5313.

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., & McCarthy, I. P. (2014). Product recovery decisions within the context of extended producer responsibility. *Journal of Engineering and Technology Management, 34*, 9–28.

Johnson, M. R., & Wang, M. H. (1998). Economical evaluation of disassembly operations for recycling, remanufacturing and reuse. *International Journal of Production Research, 36*, 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*, 309–330.

Jun, H. B., Cusin, M., Kiritsis, D., & Xirouchakis, P. (2007). A multiobjective evolutionary algorithm for EoL product recovery optimization: Turbocharger case study. *International Journal of Production Research, 45*, 4573–4594.

Jun, H. B., Lee, D. H., Kim, J. G., & Kiritsis, D. (2012). Heuristic algorithms for minimising total recovery cost of end-of-life products under quality constraints. *International Journal of Production Research, 50*, 5330–5347.

Kang, J. G., & Xirouchakis, P. (2006). Disassembly sequencing for maintenance: A survey. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 220*, 1697–1716.

Kheder, M., Trigui, M., & Aifaoui, N. (2015). Disassembly sequence planning based on a genetic algorithm. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 229*, 2281–2290.

Knowles, J.D., Thiele, L., Zitzler, E., 2006. A tutorial on the performance assessment of stochastic multiobjective optimizers. TIK-Report 214.

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

KUKA, 2020. Technical Specification, LBR iiwa 7 R800, LBR iiwa 14 R820. Technical Report. KUKA Deutschland GmbH.

Kuo, T. C. (2013). Waste electronics and electrical equipment disassembly and recycling using Petri net analysis: Considering the economic value and environmental impacts. *Computers and Industrial Engineering, 65*, 54–64.

Laili, Y., Tao, F., Pham, D. T., Wang, Y., & Zhang, L. (2019). Robotic disassembly re-planning using a two-pointer detection strategy and a superfast bees algorithm. *Robotics and Computer-Integrated Manufacturing, 59*, 130–142.

Laili, Y., Wang, Y., Fang, Y., & Pham, D. T. (2022). Solutions for robotic disassembly sequence planning with backup action. *Optimisation of Robotic Disassembly for Remanufacturing*, 111–130.

Lambert, A. (1999). Linear programming in disassembly/clustering sequence generation. *Computers & Industrial Engineering, 36*, 723–738.

Lambert, A. (2006). Exact methods in optimum disassembly sequence search for problems subject to sequence dependent costs. *Omega, 34*, 538–549.

Lambert, A. F., & Gupta, S. M. (2004). *Disassembly modeling for assembly, maintenance, reuse and recycling*. CRC Press.

Lambert, A. J. (2003). Disassembly sequencing: A survey. *International Journal of Production Research, 41*, 3721–3759.
Lee, H. B., Cho, N. W., & Hong, Y. S. (2010). A hierarchical end-of-life decision model for determining the economic levels of remanufacturing and disassembly under environmental regulations. *Journal of Cleaner Production, 18*, 1276–1283.
Li, J., Barwood, M., & Rahimifard, S. (2018). Robotic disassembly for increased recovery of strategically important materials from electrical vehicles. *Robotics and Computer-Integrated Manufacturing, 50*, 203–212.
Li, W., Xia, K., Gao, L., & Chao, K. M. (2013). Selective disassembly planning for waste electrical and electronic equipment with case studies on liquid crystal displays. *Robotics and Computer-Integrated Manufacturing, 29*, 248–260.
Liu, J., Zhou, Z., Pham, D. T., Xu, W., Cui, J., & Yang, C. (2020a). Service platform for robotic disassembly planning in remanufacturing. *Journal of Manufacturing Systems, 57*, 338–356.
Liu, J., Zhou, Z., Pham, D. T., Xu, W., Ji, C., & Liu, Q. (2018a). Robotic disassembly sequence planning using Enhanced Discrete Bees Algorithm in remanufacturing. *International Journal of Production Research, 56*, 3134–3151.
Liu, J., Zhou, Z., Pham, D. T., Xu, W., Ji, C., & Liu, Q. (2020b). Collaborative optimization of robotic disassembly sequence planning and robotic disassembly line balancing problem using improved discrete Bees algorithm in remanufacturing. *Robotics and Computer-Integrated Manufacturing, 61*, Article 101829.
Liu, J., Zhou, Z., Pham, D. T., Xu, W., Yan, J., Liu, A., et al. (2018b). An improved multi objective discrete bees algorithm for robotic disassembly line balancing problem in remanufacturing. *The International Journal of Advanced Manufacturing Technology, 97*, 3937–3962.
Lu, C., & Liu, Y. C. (2012). A disassembly sequence planning approach with an advanced immune algorithm. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 226*, 2739–2749.
Luo, Y., Peng, Q., & Gu, P. (2016). Integrated multilayer representation and ant colony search for product selective disassembly planning. *Computers in Industry, 75*, 13–26.
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*, 7007–7027.
Matsumoto, M., & Ijomah, W. (2013). Remanufacturing. In *Handbook of Sustainable Engineering* (pp. 389–408). Springer.
McGovern, S., & Gupta, S. M. (2011). *The disassembly line: Balancing and modeling.* McGraw-Hill.
Meng, K., Lou, P., Peng, X., & Prybutok, V. (2016). An improved co-evolutionary algorithm for green manufacturing by integration of recovery option selection and disassembly planning for end-of-life products. *International Journal of Production Research, 54*, 5567–5593.
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*, 72–85.
Ondemir, O., & Gupta, S. M. (2014a). A multi-criteria decision making model for advanced repair-to-order and disassembly to-order system. *European Journal of Operational Research, 233*, 408–419.
Ondemir, O., & Gupta, S. M. (2014b). Quality management in product recovery using the internet of things: An optimization approach. *Computers in Industry, 65*, 491–504.
Ong, S., Chang, M., & Nee, A. (2021). Product disassembly sequence planning: State-of-the-art, challenges, opportunities and future directions. *International Journal of Production Research*, 1–16.
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*, 3234–3243.
Pham, D. T., & Castellani, M. (2015). A comparative study of the Bees Algorithm as a tool for function optimisation. *Cogent Engineering, 2*, 1091540.
Pham, D. T., Castellani, M., & Le Thi, H. A. (2014). Nature-inspired intelligent optimisation using the bees algorithm. *Transactions on Computational Intelligence XIII. Springer*, 38–69.
Pham, D. T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., & Zaidi, M. (2006). The Bees Algorithm—a novel tool for complex optimisation problems. *Intelligent Production Machines and Systems. Elsevier*, 454–459.
Pornsing, C., & Watanasungsuit, A. (2014). Discrete particle swarm optimization for disassembly sequence planning. In *2014 IEEE International Conference on Management of Innovation and Technology* (pp. 480–485).
Priyono, A., Ijomah, W., & Bititci, U. S. (2016). Disassembly for remanufacturing: A systematic literature review, new model development and future research needs. *Journal of Industrial Engineering and Management (JIEM), 9*, 899–932.
Qu, J., Wang, W., Bai, K., & Jin, D. (2015). Guiding disassembly sequence planning based on improved fruit fly optimization algorithm. *5th International Conference on Advanced Design and Manufacturing Engineering.* Atlantis Press.
Ramírez, F. J., Aledo, J. A., Gamez, J. A., & Pham, D. T. (2020). Economic modelling of robotic disassembly in end-of-life product recovery for remanufacturing. *Computers & Industrial Engineering, 142*, Article 106339.
RIC, 2017. What is remanufacturing? http://www.remancouncil.org/educate/remanufacturing-information/what-is-remanufacturing.
Rickli, J. L., & Camelio, J. A. (2013). Multi-objective partial disassembly optimization based on sequence feasibility. *Journal of Manufacturing Systems, 32*, 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*, 7496–7512.
Shan, H., Li, S., Huang, J., Gao, Z., & Li, W. (2007). Ant colony optimization algorithm-based disassembly sequence planning. In *2007 International conference on mechatronics and automation* (pp. 867–872).
Shokohyar, S., Mansour, S., & Karimi, B. (2014). A model for integrating services and product EOL management in sustainable product service system (S-PSS). *Journal of Intelligent Manufacturing, 25*, 427–440.
Sun, Y., Yen, G. G., & Yi, Z. (2018). IGD indicator-based evolutionary algorithm for many-objective optimization problems. *IEEE Transactions on Evolutionary Computation, 23*, 173–187.
Talbi, E. G. (2009). *Metaheuristics: From design to implementation.* Wiley.
Torres, F., Puente, S., & Díaz, C. (2009). Automatic cooperative disassembly robotic system: Task planner to distribute tasks among robots. *Control Engineering Practice, 17*, 112–121.
Touzanne, F., Henrioud, J., Perrard, C., 2001. Method of disassembly sequence generation for recycling system design, in: Proceedings of the 2001 IEEE International Symposium on Assembly and Task Planning (ISATP2001). Assembly and Disassembly in the Twenty-first Century. (Cat. No. 01TH8560), IEEE. pp. 458–463.
Tovey, C. A. (2002). Tutorial on computational complexity. *Interfaces, 32*, 30–61.
Tseng, H. E., Chang, C. C., Lee, S. C., & Huang, Y. M. (2018). A block-based genetic algorithm for disassembly sequence planning. *Expert Systems with Applications, 96*, 492–505.
Tseng, H. E., Huang, Y. M., Chang, C. C., & Lee, S. C. (2020). Disassembly sequence planning using a flatworm algorithm. *Journal of Manufacturing Systems, 57*, 416–428.
UNEP, 2017. United Nations Environmental Programme. Technical Report. 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. In *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* (pp. 173–178). Springer Berlin Heidelberg.
Vongbunyong, S., Kara, S., & Pagnucco, M. (2013a). Application of cognitive robotics in disassembly of products. *CIRP Annals - Manufacturing Technology, 62*, 31–34.
Vongbunyong, S., Kara, S., & Pagnucco, M. (2013b). Basic behaviour control of the vision-based cognitive robotic disassembly automation. *Assembly Automation, 33*(1), 38–56.
Vongbunyong, S., Kara, S., & Pagnucco, M. (2015). Learning and revision in cognitive robotics disassembly automation. *Robotics and Computer-Integrated Manufacturing, 34*, 79–94.
Vongbunyong, S., Vongseela, P., & Sreerattana-Aporn, J. (2017). A process demonstration platform for product disassembly skills transfer. *Procedia CIRP, 61*, 281–286.
Wang, H., Xiang, D., Rong, Y., & Zhang, L. (2013). Intelligent disassembly planning: A review on its fundamental methodology. *Assembly Automation*.
Wang, K., Gao, L., Li, X., & Li, P. (2021). Energy-efficient robotic parallel disassembly sequence planning for end-of-life products. *IEEE Transactions on Automation Science and Engineering*.
Wang, L., Wang, X. V., Gao, L., & Váncza, J. (2014). A cloud-based approach for WEEE remanufacturing. *CIRP annals, 63*, 409–412.
Xia, K., Gao, L., Li, W., & Chao, K. M. (2014). Disassembly sequence planning using a Simplified Teaching- Learning-Based Optimization algorithm. *Advanced Engineering Informatics, 28*, 518–527.
Xu, W., Tang, Q., Liu, J., Liu, Z., Zhou, Z., & Pham, D. T. (2020). Disassembly sequence planning using discrete bees algorithm for human-robot collaboration in remanufacturing. *Robotics and Computer- Integrated Manufacturing, 62*, Article 101860.
Yuce, B., Packianather, M. S., Mastrocinque, E., Pham, D. T., & Lambiase, A. (2013). Honey bees inspired optimization method: The bees algorithm. *Insects, 4*, 646–662.
Zhang, Z. F., Feng, Y. X., Tan, J. R., Jia, W. Q., & Yi, G. D. (2015). A novel approach for parallel disassembly design based on a hybrid fuzzy-time model. *Journal of Zhejiang University-Science A, 16*, 724–736.
Zhong, L., Youchao, S., Gabriel, O. E., & Haiqiao, W. (2011). Disassembly sequence planning for maintenance based on metaheuristic method. *Aircraft Engineering and Aerospace Technology*.
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.
Zitzler, E., Knowles, J., & Thiele, L. (2008). Quality assessment of Pareto set approximations. *Multiobjective Optimization. Springer*, 373–404.
Zitzler, E., & Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International conference on parallel problem solving from nature* (pp. 292–301). Springer.