Evolutionary many-objective optimization for mixed-model disassembly line balancing with multi-robotic workstations

Fang, Yilin; Liu, Quan; Li, Miqing; Laili, Yuanjun; Pham, Duc Truong

DOI:

[10.1016/j.ejor.2018.12.035](https://doi.org/10.1016/j.ejor.2018.12.035)

License:

Creative Commons: Attribution-NonCommercial-NoDerivs (CC BY-NC-ND)

Document Version

Peer reviewed version

Citation for published version (Harvard):

Fang, Y, Liu, Q, Li, M, Laili, Y & Pham, DT 2018, 'Evolutionary many-objective optimization for mixed-model disassembly line balancing with multi-robotic workstations', *European Journal of Operational Research*.
<https://doi.org/10.1016/j.ejor.2018.12.035>

[Link to publication on Research at Birmingham portal](#)

General rights

Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

- Users may freely distribute the URL that is used to identify this publication.
- Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
- User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
- Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy

While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Accepted Manuscript

Evolutionary many-objective optimization for mixed-model
disassembly line balancing with multi-robotic workstations

Yilin Fang, Quan Liu, Miqing Li, Yuanjun Laili, Duc Truong Pham

PII: S0377-2217(18)31108-1
DOI: <https://doi.org/10.1016/j.ejor.2018.12.035>
Reference: EOR 15559

To appear in: *European Journal of Operational Research*

Received date: 24 April 2018
Revised date: 29 September 2018
Accepted date: 20 December 2018

Please cite this article as: Yilin Fang, Quan Liu, Miqing Li, Yuanjun Laili, Duc Truong Pham, Evolutionary many-objective optimization for mixed-model disassembly line balancing with multi-robotic workstations, *European Journal of Operational Research* (2018), doi: <https://doi.org/10.1016/j.ejor.2018.12.035>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Highlights

- Mixed-model disassembly line balancing with multi-robotic workstations is studied.
- The properties of subassembly dependent task and parallel task are studied.
- A mathematical model for our problem with many objectives is developed.
- A problem knowledge-leveraging many-objective evolutionary algorithm is proposed.

Evolutionary many-objective optimization for mixed-model disassembly line balancing with multi-robotic workstations

Yilin Fang^{a,b,*}, Quan Liu^{a,b}, Miqing Li^c, Yuanjun Laili^d, Duc Truong Pham^e

^a*School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China*

^b*Key Laboratory of Fiber Optic Sensing Technology and Information Processing, Ministry of Education, Wuhan 430070, China*

^c*CERCIA, School of Computer Science, University of Birmingham, Birmingham, B15 2TT, UK*

^d*School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China*

^e*School of Mechanical Engineering, University of Birmingham, Birmingham, B15 2TT, UK*

Abstract

In the remanufacturing industries, automated disassembly becomes one of the most promising solution in achieving economic benefit. Robotic disassembly line balancing is a key problem that enables automated disassembly to be implemented at industrial scale. This paper focuses on evolutionary many-objective optimization for mixed-model disassembly line balancing with multi-robotic workstations. In each workstation, multiple skilled robots perform different tasks belonging to the different end-of-life products or subassemblies simultaneously. Based on the transformed AND/OR graph and parallel disassembly, a mathematical programming model is proposed to minimize the cycle time, the total energy consumption, the peak workstation energy consumption, and the number of robots being used simultaneously. Furthermore, a problem knowledge-leveraging evolutionary algorithm, including encoding/decoding scheme, initialization approach and problem-specific variation operators, is developed to deal with the above problem. Comprehensive experiments are conducted based on 8 product models and 63 problem instances generated in this study. In particular, a comparative study of our proposed algorithm and 5 representative evolutionary algorithms selected from the 3 classes of approaches of dealing with many-objective problems are provided. Then some insights with respect to the design of evolutionary algorithms for our problem are gained from the investigation.

Keywords: Metaheuristics, Disassembly line balancing, Multi-robotic workstation, Many-objective optimization, Evolutionary algorithm

*Corresponding author.

Email addresses: fangspirit@whut.edu.cn (Yilin Fang), quanliu@whut.edu.cn (Quan Liu), limitsing@gmail.com (Miqing Li), llyj0721@gmail.com (Yuanjun Laili), d.t.pham@bham.ac.uk (Duc Truong Pham)

1. Introduction

Disassembly, as one of the key steps in the treatment of end-of-life (EOL) products, involves the extraction of parts (**components**), subassemblies and materials from returned products to be remanufactured (**McGovern & Gupta, 2007**). It has **attracted** increasing attention over the past few decades, since it enables the recovery of embodied value left within disposed EOL products, and can be profitable and less harmful to the environment. The disassembly line, which consists of a number of serial workstations, is an appropriate layout for disassembly operations due to its **suitability to automation** and high productivity compared with single workstation and disassembly cell (Ilgin et al., 2017). A disassembly line must be balanced so that it works as efficiently as possible, and the disassembly line balancing problem (DLBP) is the process of allocating a set of disassembly tasks to an ordered sequence of workstations in such a way that precedence relations of EOL product model are not violated and some performance measures are optimized (**Güngör & Gupta, 2002**).

Although the disassembly line has been extensively studied in the academic literature, it has generally been neglected in industry practice. One main reason for this is that the operations tend to be manually executed, thus resulting in high costs. Currently, **many** attempts have been made to automate the disassembly operations. Though some associated technical challenges need to first be addressed, the automated disassembly line is the most promising approach to increase the economic feasibility of the disassembly (Vongbunyong & Chen, 2015). **In addition**, as the complexity and variability of the EOL products coming into disassembly lines become higher, more complex problems arise. On the one hand, most of the EOL products have **dozens** of disassembly tasks. **If they are disassembled in a disassembly line** having only one robot in each workstation, hundreds of workstations will need. In fact, a team-oriented system (**Dimitriadis, 2006**) can be adopted to provide several advantages over traditional **lines** such as shorter line length, shorter cycle time and lower material handling costs, where **multiple** robots simultaneously perform different tasks on the same workstation. On the other hand, traditional disassembly lines are mostly designed for the case of a single product **model**. This single-model line is quite limited that different EOL product models cannot be handled simultaneously and continuously on the same disassembly line. In contrast, mixed-model line, in which two or more product models are processed simultaneously, is more suitable to improve the flexibility to adapt to **different product models** (Paksoy et al., 2013), satisfies **variant** disassembly demand, and reduces **line** building and maintaining costs. In view of these advantages, it is necessary to take the characteristics of automated disassembly, team-oriented system and mixed-model line into consideration in the design of disassembly lines since they are more convenient for modern **industry's** needs.

There are about 120 papers on DLBP area from 1999 (introduction of DLBP) to 2018, which can

be classified based on several factors such as line design, station type and task properties. Detailed explanations and classifications of disassembly line design and balancing can be found in Özceylan et al. (2018), Bentaha et al. (2015a) and Battaia & Dolgui (2013). However, their reviews show that few papers have studied DLBP combining the aforementioned characteristics. These relevant works are summarized in Table 1, differentiated with respect to several features of our study. It is observed that (i) there are only four studies dealing with mixed-model DLBP and the sole study dealing with robotic DLBP, (ii) no publication has treated team-oriented disassembly line system. To bridge the gap, this study presents a multi-robotic workstation approach for designing mixed-model disassembly lines, and considers a new mixed-model disassembly line balancing with multi-robotic workstations (MDLB-MR), depicted in Fig. 1. MDLB-MR is a new type of generalized DLBP in which there is a possibility of assigning more different skilled robots than one to each workstation according to the features of different EOL product models, and is designed to disassemble mixed-model EOL products. When extending the literature to assembly domain, it is revealed that there are two assembly line balancing problems with most similar characteristics are as follows: two-sided assembly line balancing problem (TALBP) (Bartholdi, 1993) and multi-manned assembly line balancing problem (MALBP) (Dimitriadis, 2006). MDLB-MR differs from these two problems in several respects. First of all, there are no positional and zoning constraints in MDLB-MR rather than TALBP. It is possible to assign more than two robots to each workstation. Secondly, the main difference between MDLB-MR and MALBP is that all the workers have identical skills and perform that tasks with identical speeds in MALBP. However, MDLB-MR incorporate many skilled robots that can perform the tasks with different operation times. Moreover, a group of workers simultaneously perform different tasks on the same individual product in MALBP, while only a group of tasks belonging to the different EOL products or subassemblies can be performed simultaneously by a group of robots in MDLB-MR.

Clearly, one of the most significant features of MDLB-MR is robot paralleling in each workstation. Using robot paralleling in disassembly lines contributes to the recovery processes positively in terms of high productivity, higher recovery rates, short cycle times and reduction of station idle time. In order to obtain these potential benefits of parallel disassembly, appropriate task precedence relation representation scheme should be adopted to distinguish the tasks from different subassemblies and provide the information on which tasks can be performed simultaneously. The most commonly used representation scheme in DLBP literature is task precedence diagram (TPD). The classical AND TPDs originate from traditional assembly line balancing problem (ALBP) (Boysen et al., 2007; Bullinger & Ammer, 1984). In AND TPDs, the task precedence relationships are limited to simple AND types. Güngör & Gupta (2001, 2002) pointed out that OR precedence and complex AND/OR precedence relationships should be considered

in the disassembly case. Furthermore, two types of AND/OR-graph-based TPDs, AND/OR-graph-based disassembly precedence diagram (Altekin et al., 2008; Altekin & Akkan, 2012) and transformed AND/OR graph (TAOG) (Koc et al., 2009; Bentaha et al., 2018), were proposed to model all possible disassembly alternatives of an EOL product. Both of them take OR successor relationship into account. However, in contrast to AND/OR-graph-based disassembly precedence diagram constituted of tasks, TAOG fully separates the definition of task and subassembly. The tasks in TAOG are defined based on not only the contacts but also the current subassembly. In addition, in TAOG, not only sequential tasks but also parallel tasks, which take part a subassembly into two, each of which has at least two parts, can be considered. Hence TAOG can distinguish the tasks from different subassemblies clearly and provides the information on which tasks can be performed simultaneously, thereby representing parallel disassembly in a more natural and straightforward way. For these reasons, TAOG is used in this study. The properties of subassembly dependent (SD) task and parallel task in TAOG are also considered to study MDLB-MR. Few studies in DLBP literature, however, have taken into account parallel disassembly and used TAOG, which are listed in Table 1. As shown in Table 1, to the best of our knowledge, only two DLBP studies, Aydemir-Karadag & Turkbey (2013) and Hezer & Kara (2015), considered the idea of parallel disassembly. Nevertheless, in contrast to the robot paralleling introduced in our study, station paralleling is used to treat the case of stochastic DLBP in Aydemir-Karadag & Turkbey (2013) and line paralleling is adopted in Hezer & Kara (2015).

Table 1

Relevant literature on DLBP area.

Features	Publications
Mixed-model DLBP	Agrawal & Tiwari (2008); Paksoy et al. (2013); Riggs et al. (2015); Ilgin et al. (2017)
Robotic DLBP	Lin et al. (2018)
TAOG-based precedence relation representation	Koc et al. (2009); Aydemir-Karadag & Turkbey (2013); Özceylan & Paksoy (2013, 2014); Paksoy et al. (2013); Bentaha et al. (2015b,c, 2018); Hezer & Kara (2015); Mete et al. (2016, 2018); Zheng et al. (2018)
Considering parallel disassembly	Aydemir-Karadag & Turkbey (2013); Hezer & Kara (2015)
Pareto-based multi-objective optimization	Ding et al. (2010); Aydemir-Karadag & Turkbey (2013); Kalayci et al. (2015); Jiang et al. (2016); Xiao et al. (2017); Zhang et al. (2017); Gao et al. (2018); Liu et al. (2018); Pistolesi et al. (2018); Zhu et al. (2018)

The objective of DLBP is to utilize the resources as efficiently as possible while meeting the demand. There are some frequently used objectives for DLBP such as cycle time, number of workstations, profit,

and hazardous content (Mete et al., 2016; Jiang et al., 2016; Özceylan et al., 2018). Moreover, in MDLB-MR, robots are scarce resources and the robot energy consumption is a major expense. The reduction of energy consumption beneficially influences industry's impact on the environment (Nilakantan et al., 2017). In reality, there are usually more than three objectives in complex DLBPs such as MDLB-MR, which conflict with each other normally and need to be addressed simultaneously (Paksoy et al., 2013). In our study, four objectives, include the cycle time, the total energy consumption, the peak workstation energy consumption and the number of robots being used, are considered simultaneously. Therefore, it is necessary to investigate many-objective DLBPs, which is an essential class of multi-objective DLBPs with at least four objectives (Giagkiozis & Fleming, 2015; Li et al., 2015), to achieve the equilibrium among all the objectives.

Due to the DLBP's NP-hard nature (McGovern & Gupta, 2007) and the inclusion of simultaneously considered multiple objectives listed above, exact solution approaches are not suitable for large-scale multi-objective DLBPs. In DLBP literature, there is an increasing trend for using metaheuristics which can be divided into three classes (Battaia & Dolgui, 2013): (i) neighborhood methods such as tabu search algorithms (Kalayci & Gupta, 2014) and simulated annealing algorithms (Prakash et al., 2012), (ii) evolutionary approaches such as genetic algorithms (McGovern & Gupta, 2007; Aydemir-Karadag & Turkbey, 2013; Kalayci et al., 2016; Jiang et al., 2016; Pistoiesi et al., 2018), and (iii) swarm intelligence based methods such as ant colony algorithms (Agrawal & Tiwari, 2008; Ding et al., 2010; Mete et al., 2018), bee algorithms (Liu et al., 2018), artificial bee colony algorithms (Kalayci et al., 2015; Gao et al., 2018), particle swarm optimization algorithms (Kalayci & Gupta, 2013; Xiao et al., 2017), artificial fish swarm algorithms (Zhang et al., 2017), gravitational search algorithms (Y. et al., 2017), and firefly algorithms (Zhu et al., 2018). Some of these studies deal with multi-objective DLBPs, but evaluate the solution performance by making aggregation or by using lexicographic approach, in which the solution is very sensitive to weights or orders (Battaia & Dolgui, 2013). In contrast, Pareto-based metaheuristics overcome the disadvantage as they provide a uniform set of Pareto-optimal solutions per run, with no weights or orders to specify (Pistoiesi et al., 2018). There are some DLBP publications where an approximation of Pareto front is constructed by using Pareto-based metaheuristics, which are listed in Table 1. However, Pareto-based metaheuristics also encounter difficulties in many-objective DLBPs because Pareto-dominance will gradually lack enough selection pressure to pull the population toward Pareto front with the number of objectives increasing. Another problem is that it is unclear how to preserve diversity in solutions (Giagkiozis & Fleming, 2015). One way to overcome these drawbacks is to blend non-Pareto evolution and Pareto evolution in a Pareto-based metaheuristic (Li et al., 2016a). Based on this idea, we address evolutionary approaches for MDLB-MR and present a many-objective evolutionary algorithm (EA) to strike a

balance between convergence and diversity in the objective space by combining non-Pareto evolution with Pareto evolution. To the best of authors' knowledge, there is no multi-objective metaheuristic reported in DLBP literature to address this issue.

As mentioned above, this paper deals with MDLB-MR based on TAOG and allows parallel disassembly by using robot paralleling. A mathematical formulation and a many-objective EA based on our bi-criterion evolution (BCE) framework (Li et al., 2016a) are proposed for MDLB-MR to choose a disassembly alternative and assign the corresponding disassembly tasks to opened parallel robots in each workstation. The near Pareto-optimal solutions can be obtained by using the developed algorithm for many-objective optimization of MDLB-MR. Moreover, a new set of test instances is generated based on 8 TAOG models of realistic EOL products from the literature, and the search ability of the proposed algorithm is compared with some representative EAs in term of hyper-volume (HV) (Zitzler & Thiele, 1999). Then some insights with respect to the design of EAs for MDLB-MR are gained. Compared with the existing studies in DLBP, the differences and main contributions of this study can be summarized by the following four respects.

- (1) Our research is the first attempt to discuss MDLB-MR, which is more practical and generalized than traditional DLBPs and very well applicable to complex mixed-products disassembly shop with automated disassembly lines, but is rarely mentioned in the literature.
- (2) In MDLB-MR, one or more robots can be assigned to a workstation and perform tasks simultaneously. To obtain the benefits of robot paralleling, TAOG is used to model the precedence relations among tasks. The properties of subassembly dependent task and parallel task are also considered to study MDLB-MR.
- (3) We develop a mathematical model for MDLB-MR with four objectives, for a fixed number of workstations, which can be used to solve small-sized problems optimally.
- (4) A problem-specific bi-criterion evolutionary algorithm (PBEA), which focuses on many-objective optimization of MDLB-MR, is developed to obtain Pareto-optimal solutions for medium-sized and large-sized problems.

The remainder of the paper is organized as follows. In section 2, the concept of mixed-model disassembly line with multi-robotic workstations (MDL-MR) is introduced, along with detailed definition of MDLB-MR. Section 2 also provides a small-sized explanatory example in order to illustrate the superiority of task precedence relations modeling based on TAOG and parallel SD tasks. The notations and the mathematical model are presented in Section 3. Section 4 describes the implementation details of PBEA. Afterward, experimental studies are provided in Section 5. Finally, Section 6 draws the conclusion and highlights possible future research avenues.

2. Description of the mixed-model disassembly line balancing with multi-robotic workstations

2.1. Problem definition and assumptions

To demonstrate the intricacies of practical automated disassembly procedure, serial paced MDL-MRs and one of their balancing problems are addressed. The main difference between this class of disassembly lines and classical simple disassembly lines is the number of robots that can be allocated to each workstation. In MDL-MR, a set of EOL product models can be simultaneously disassembled and each product model has its own set of precedence relations. At each workstation, more than one robots are installed in such a way that the robots can perform simultaneously. At any point during cycle time, robots assigned to the same workstation can simultaneously perform a group of disassembly tasks waiting at the workstation, with these tasks belonging to the different EOL products or subassemblies. Due to characteristics of product and workstation such as product size, tools availability and workstation construction, the maximum number of robots which can be allocated to the same workstation is predetermined by the line system designer (Fattahi et al., 2011). Moreover, different from multi-manned assembly line with identical skilled workers, all robots in MDL-MR have different skills and process the assigned tasks with the different operation periods. For better understanding MDL-MRs, we use Fig. 1 to illustrate a MDL-MR where there are 3 workstations, 6 different skilled robots, and 2 EOL product models.

Fig. 1. An example of the MDL-MR layout.

MDL-MR consists of the assignment of the chosen disassembly tasks of several different types of EOL products to a set of available robots and the allocation of these robots to an ordered sequence of workstations, such that task precedence relations of each EOL product are not violated and some objectives are optimized. In this study, these objectives include ($F1$) the minimization of cycle time, ($F2$) the minimization of total energy consumption, ($F3$) the minimization of peak workstation energy consumption and ($F4$) the minimization of number of robots being used, and we assume that the number

of workstations is predetermined. To some extent, MDLB-MR can be viewed as an extension of traditional type II DLBP which aims to minimize cycle time (hence maximize *disassembly* rate) given a fixed number of workstations.

The following assumptions are given for MDLB-MR which are similar to those mentioned in Koc et al. (2009), Li et al. (2016b), Nilakantan et al. (2017) and Roshani & Giglio (2017).

- (1) A serial paced disassembly line system is configured for the mass-disassembly of multiple mixed-model products and the all feasible precedence relations between the subassemblies and the disassembly tasks of each model are given.
- (2) Each disassembly takes apart the product or subassembly into two new subassemblies and each product undergoes complete disassembly.
- (3) Both sequential SD tasks and parallel SD tasks in TAOG are considered to fulfill parallel disassembly. Note that references to tasks are to sequential or parallel SD tasks in the rest of this paper, unless otherwise stated.
- (4) Time taken to perform a disassembly task depends on the type of robot assigned and it is deterministic, constant, and discrete.
- (5) Operation/standby energy consumptions of each robot per time unit are given.
- (6) Workstations are aligned in a serial manner and the number of workstations is given.
- (7) More than one robot can be permitted to allocate to each workstation and the maximum number of robots in each workstation is given.
- (8) Any robot can perform only one disassembly task at the same time and each task must be processed by exactly one robot.
- (9) Disassembly tasks cannot be split and the precedence relationship in TAOG should be met while assigning the tasks.
- (10) Workstation processing time has to be less than the cycle time.
- (11) Setup times between disassembly tasks are ignored and no work-in-process inventory buffer is considered.

2.2. *Disassembly precedence relations*

In this study, TPD developed by Koc et al. (2009) is employed to model all possible alternatives for the disassembly process and precedence relationships among subassemblies and tasks. It is called TAOG and based on AND/OR graph. An AND/OR graph depicts all possible ways of the complete disassembly of a product into its basic parts and shows all possible subassemblies. TAOG is a modified version of an AND/OR graph with explicit information on the precedence relations between tasks. A

detailed explanation of AND/OR graph and TAOG can be found in Homem & Sanderson (1990) and Koc et al. (2009), respectively.

The use of TAOGs is illustrated in Appendix A. Fig. A2 shows the TAOGs of the top 2 EOL product models used in this paper. The TAOG model 1 in Fig. A2(a) corresponds to a sample product example (Koc et al., 2009) and is constructed as described below. Each subassembly is represented by an artificial node labelled A_a^m , where $a \in A^m$ and $m \in M$. Set M and A^m contain the indices of all TAOG models and all possible subassemblies that can be generated by the tasks from B^m , respectively. The set of parts in a subassembly is denoted by a number sequence below the corresponding artificial node. For simplicity, subassemblies with only one part are not shown. For instance, artificial node A_0^1 represents the sample product which can be noted as '1/7', A_1^1 represents the subassembly '1/4, 6, 7' which consists of part 1 through 4, part 6 and 7, etc. (see Fig. A2(a)). Each task is symbolized by a normal node labelled B_n^m , where $n \in B^m$ and $m \in M$. Set B^m contains the indices of all the possible tasks that can be performed on the m^{th} model. An artificial node may be preceded or succeeded by more than one normal node; however, only one of the predecessors and one of the successors can be selected during disassembly. Arcs in the TAOG are of two types: AND type and OR type. While the AND-type arc imposes a mandatory precedence relation case, the OR-type arc allows alternative selection of the following routes. An indicator of OR-type arc is shown by a small curve.

In addition, A disassembly tree (DT) in TAOG model m represents one way of completely disassembling the corresponding EOL product and has $|P^m| - 1$ tasks where P^m and $|P^m|$ denote the set of part indices and the number of parts in the model, respectively. There could be many DTs in model m but only one of them can be selected to disassemble the product.

2.3. Illustrative example

To give a better insight of MDLB-MR, a small-scale instance of MDL-MR is considered where there are 3 multi-robotic workstations ($|J| = 3$) and 10 robots ($R = \{r_1 \dots r_{10}\}$, $|R| = 10$). It is possible to assign up to 3 robots from the 10 robots to each workstation ($R_{max} = 3$). There are 2 EOL product models ($|M| = 2$) to be simultaneously disassembled, with model 1 and model 2 constructed with 7 and 10 parts respectively ($|P^1| = 7$, $|P^2| = 10$). To illustrate the superiority of using TAOG, the classical AND TPD is selected as the competitor. The disassembly precedence relations of these 2 EOL product models are represented based on TAOGs and AND TPDs, respectively. These resulting graphs are shown in Appendix A. In addition, Appendix B depicts referenced processing times for SD tasks in the two TAOGs given in Fig. A2. Based on these data, the processing times for all SD tasks in TAOGs by different robots are randomly generated between $[t_{bro} \cdot 0.7, t_{bro} \cdot 1.3]$ and t_{bro} is the referenced processing time. Note that, AND TPD not containing the information on which subassembly the relevant sequential

subassembly independent (SI) task applies, the processing time of an sequential SI task in AND TPD needs to be set as the maximum of the processing times of the corresponding SD tasks in TAOG (Koc et al., 2009).

For the sake of simplicity, the objective is to minimize the cycle time by balancing this small-scale instance of MDL-MR. Note that the motivation behind this example is to illustrate the following 3 properties which can be utilized to solve MDLB-MR.

- (1) All possible DTs which are generated from a TAOG instead of partial DTs provided by an AND TPD can be scanned to optimize MDLB-MR.
- (2) The accurate processing time of each sequential/parallel SD task in TAOG instead of the unified maximum processing time of each sequential SI task in AND TPD can be considered to balance MDL-MR.
- (3) Containing the full information on which subassemblies the SD tasks apply, the SD tasks belonging to the different subassemblies in the same model can be assigned to different robots in the same workstation, processed in parallel.

Fig. 2. Optimal solutions for an example of MDLB-MR based on TAOGs and AND TPDs, solved by LINGO.

The example of MDLB-MR has been solved through 2 similar single-objective modified versions of the mathematical model that is presented in Section 3.2, coded by LINGO 11.0 optimizer so as to obtain optimal solutions based on TAOGs and AND TPDs, respectively. Fig. 2 illustrates the optimal solutions of the example of MDLB-MR when the cycle time is the performance index to be minimized in the

objective function. In Fig. 2, the **SD** and **sequential SI** tasks that are assigned to each robot are indicated by white rectangles where the tasks processing times are illuminated below, and idle times of robots are showed by gray rectangles. As we can see in the bottom graph of Fig. 2, the optimal cycle time is 82 time units when **AND TPDs** are utilized as the task precedence model for the example of MDLB-MR. On the other hand, if we calculate the optimal solution based on TAOGs and properties of **SD** tasks for the same problem we obtain the solution given in the upper-left graph of Fig. 2, and the optimal cycle time drops to 31 time units significantly. As can be observed, robots are better used, and less idle times occur. This **example** shows the remarkable superiority of using TAOG and **SD** tasks over **AND TPD** and **sequential SI** tasks, partly because all possible DTs and the accurate operation time of each **SD task** can be utilized to search optimal solution, and also partly because **SD tasks** in the same product can be processed in parallel while **sequential SI tasks** must be performed one by one. In this study, all the 3 properties of TAOG and **SD task** will be fully utilized to address MDLB-MR. Note that, the first 2 properties are not uncommon in DLBP literature (Koc et al., 2009; Paksoy et al., 2013; Aydemir-Karadag & Turkbey, 2013; Mete et al., 2016). However, the third property is one of our contributions in this study. To the best of our knowledge, this property has not been exploited previously in **DLBP** literature.

3. Mathematical formulation

The mathematical **model for MDLB-MR** is presented in this section. **Some constraints** in this model are those deriving from the models in Paksoy et al. (2013), Li et al. (2016b) and Nilakantan et al. (2017).

3.1. Notations

The indices, sets and parameters and variables used in the mathematical formulations are given in Tables 2, 3 and 4, respectively.

Table 2

The indices used in the mathematical model.

Notation	Definition	Notation	Definition
n, h	Index of normal nodes	r	Index of robots
a	Index of artificial nodes	j, v	Index of workstations
m, m'	Index of TAOG models	$i(B_n^m)$	Index of B_n^m in D

Table 3

The sets and parameters used in the mathematical model.

Notation	Definition	Notation	Definition
M	Set of TAOG model indices	$P_{all}(B_n^m)$	Set of all normal predecessors of B_n^m in D

(continued on next page)

Table 3 (continued)

Notation	Definition	Notation	Definition
P^m	Set of part indices in model m	$S_{all}(B_n^m)$	Set of all normal successors of B_n^m in D
B^m	Set of normal node indices in model m	B_n^m	Task n of model m
A^m	Set of artificial node indices in model m	A_a^m	Artificial node a of model m
R	Set of robot indices	ψ	A big positive number
J	Set of workstation indices	R_{max}	Maximum number of robots in each workstation
$P(A_a^m)$	Set of immediate predecessor of A_a^m in model m	OE_r	Operation energy consumption of r^{th} robot per time unit
$S(A_a^m)$	Set of immediate successor of A_a^m in model m	SE_r	Standby energy consumption of r^{th} robot per time unit
D	Set of all selected tasks for disassembly	t_{mnr}	Processing time of task B_n^m by r^{th} robot

Table 4

The variables used in the mathematical model.

Notation	Definition	Notation	Definition
X_{mnrj}	1, if B_n^m is assigned to r^{th} robot in j^{th} workstation; 0, otherwise	$Y_{mnm'h}$	1, if B_n^m starts its execution before $B_h^{m'}$ in the same robot; 0, otherwise
W_{rj}	1, if r^{th} robot is used in j^{th} workstation; 0, otherwise	S_{mn}	Starting time of task B_n^m
Z_{mn}	1, if B_n^m is selected to perform; 0, otherwise		

3.2. Mathematical model for MDLB-MR

The objective functions are demonstrated in Table 5. The first and the second column provide an index and a mathematical representation for each objective which is used in the model, respectively.

Table 5

The objective functions of the mathematical model.

Index	Formula
$F1$	$\text{MIN } CT$
$F2$	$\text{MIN } TE = \sum_{j \in J} (\sum_{r \in R} \sum_{m \in M} \sum_{n \in B^m} OE_r \cdot t_{mnr} \cdot X_{mnrj} + \sum_{r \in R} SE_r (CT \cdot W_{rj} - \sum_{m \in M} \sum_{n \in B^m} t_{mnr} \cdot X_{mnrj}))$
$F3$	$\text{MIN } PE = \max_{j \in J} (\sum_{r \in R} \sum_{m \in M} \sum_{n \in B^m} OE_r \cdot t_{mnr} \cdot X_{mnrj} + \sum_{r \in R} SE_r (CT \cdot W_{rj} - \sum_{m \in M} \sum_{n \in B^m} t_{mnr} \cdot X_{mnrj}))$
$F4$	$\text{MIN } NR = \sum_{j \in J} \sum_{r \in R} W_{rj}$

Table 6

The constraints of the mathematical model.

Classification	Index	Formula
Cycle time constraint	C1	$S_{mn} \cdot Z_{mn} + \sum_{j \in J} \sum_{r \in R} t_{mnr} \cdot X_{mnrj} \leq CT, \forall m \in M, n \in B^m$
Occurrence straint	con- C2	$\sum_{B_n^m \in S(A_a^m)} Z_{mn} = 1, \forall m \in M, a = 0, a \in A^m$
	C3	$\sum_{B_n^m \in S(A_a^m)} Z_{mn} = \sum_{B_n^m \in P(A_a^m)} Z_{mn}, \forall m \in M, a \neq 0, a \in A^m$
	C4	$\sum_{j \in J} \sum_{r \in R} X_{mnrj} = Z_{mn}, \forall m \in M, n \in B^m$
Precedence straint	con- C5	$\sum_{B_n^m \in P(A_a^m)} \sum_{j \in J} \sum_{r \in R} X_{mnrj} \geq \sum_{B_n^m \in S(A_a^m)} \sum_{r \in R} X_{mnrj}, \forall m \in M, a \neq 0, a \in A^m, v \in J$
	C6	$\sum_{B_n^m \in P(A_a^m)} \sum_{j \in J} \sum_{r \in R} j \cdot X_{mnrj} \leq \sum_{B_n^m \in S(A_a^m)} \sum_{j \in J} \sum_{r \in R} j \cdot X_{mnrj}, \forall m \in M, a \neq 0, a \in A^m$
Sequence-related constraint	C7	$S_{mn} - S_{mh} + \psi(1 - \sum_{r \in R} X_{mhrj}) + \psi(1 - \sum_{r \in R} X_{mnrj}) \geq \sum_{r \in R} t_{mhr} \cdot X_{mhrj}$ $, \forall j \in J, m \in M, a \neq 0, a \in A^m, B_h^m \in D \cap P(A_a^m), B_n^m \in D \cap S(A_a^m)$
	C8	$S_{m'h} - S_{mn} + \psi(1 - X_{m'hrj}) + \psi(1 - X_{mnrj}) + \psi(1 - Y_{mnm'h}) \geq t_{mnr} \cdot X_{mnrj}$ $, \forall B_n^m \in D, B_h^{m'} \in D - P_{all}(B_n^m) \cup S_{all}(B_n^m), i(B_n^m) < i(B_h^{m'}), j \in J, r \in R$
	C9	$S_{mn} - S_{m'h} + \psi(1 - X_{m'hrj}) + \psi(1 - X_{mnrj}) + \psi \cdot Y_{mnm'h} \geq t_{m'hr} \cdot X_{m'hrj}$ $, \forall B_n^m \in D, B_h^{m'} \in D - P_{all}(B_n^m) \cup S_{all}(B_n^m), i(B_n^m) < i(B_h^{m'}), j \in J, r \in R$
	C10	$D = \{B_n^m Z_{mn} = 1, m \in M, n \in B^m\}, D = \sum_{m \in M} (P^m - 1)$
	C11	$P_{all}(B_n^m) \subset D, S_{all}(B_n^m) \subset D, \forall m \in M, n \in B^m$
Variable constraint	C12	$\sum_{m \in M} \sum_{n \in B^m} X_{mnrj} \leq D \cdot W_{rj}, \forall j \in J, r \in R$
	C13	$1 \leq \sum_{r \in R} W_{rj} \leq R_{max}, \forall j \in J$
	C14	$\sum_{j \in J} W_{rj} \leq 1, \forall r \in R$
	C15	$S_{mn} \geq 0, \forall m \in M, n \in B^m$
	C16	$X_{mnrj} \in \{0, 1\}, \forall m \in M, n \in B^m, j \in J, r \in R$
	C17	$W_{rj} \in \{0, 1\}, \forall j \in J, r \in R$
	C18	$Y_{mnm'h} \in \{0, 1\}, \forall B_n^m \in D, B_h^{m'} \in D - P_{all}(B_n^m) \cup S_{all}(B_n^m)$
	C19	$Z_{mn} \in \{0, 1\}, \forall m \in M, n \in B^m$

$F1$ defines to minimize the cycle time. In this model, the number of multi-robotic workstation is given and the goal is to minimize the cycle time and other objectives. $F2$ and $F3$ mean to minimize the total energy consumption and the peak workstation energy consumption, respectively. When robots march on disassembly, energy management and environmental issue become important factors for promoting green practices. This model focus on the carbon footprint caused by robot energy consumption, which includes operation and standby consumption, in the disassembly line. Also, robot can be viewed as a type of scarce

resource that is expensive and limited and must be available for disassembly. Therefore, $F4$ is designated to minimize the total number of robots being used.

The constraints used in MDLB-MR are highlighted in Table 6. The first column gives the classification that the constraint belongs to. The second and the third column provide an index and a formula for each constraint, respectively.

$C1$ defines the cycle time constraint. $C2$ and $C3$ guarantee that only a disassembly tree can be selected to completely disassembly the product of a TAOG model, and $C4$ ensures that a selected task is assigned to one of the robots in one of the workstations. $C5$ and $C6$ check the precedence constraints of the selected tasks. $C7$, $C8$ and $C9$ control the sequence-dependent starting time of each selected task. For a pair of selected tasks B_n^m and B_h^m , if task B_h^m is an immediate predecessor of task B_n^m on a same workstation, then $C7$ becomes active. If a pair of selected tasks has no precedence relations and they are allocated to a same robot in a same workstation, $C8$ and $C9$ become active. $C10$ and $C11$ are supplements to $C8$ and $C9$. $C10$ defines the set of all selected tasks, which is denoted by D , and $C11$ defines the set of all predecessors and the set of all successors of a selected task in D . $C12$ confines that, if a selected task is assigned to the r^{th} robot of the j^{th} workstation, then $W_{r,j}$ must be equal 1 (since the r^{th} robot should be utilized in the j^{th} workstation). $C13$ states that, at least one robot can be permitted to allocate to each workstation while the maximum number of robots in each workstation is R_{max} . $C14$ enforces that each selected robot can be assigned to only one workstation. Moreover, there are also some other variable constraints to determine which variables are feasible for MDLB-MR, which are denoted by $C15 - 19$ in Table 6.

4. Problem-specific bi-criterion evolutionary algorithm for MDLB-MR

In this study, PBEA is developed to solve MDLB-MR. The framework of the proposed PBEA originates from our original BCE framework (Li et al., 2016a), which is depicted in Algorithm 1. The detail of PBEA are explained in this section.

4.1. Procedure of PBEA

PBEA consists of two evolution parts which are based on non-Pareto criterion (NPC) and Pareto criterion (PC) respectively. Two evolutionary populations in PBEA, NPC population and PC population, evolve collaboratively with an abundant information exchange to enhance the convergence of PC population and compensate the diversity loss of NPC population. The procedure of PBEA starts with generating an NPC population for initialization using the solution representation in Section 4.2 and problem-specific initialization procedure in Section 4.3. Initializations of the reference point and the PC population are performed in turn. Then, steps 2.1–2.3 are performed in a cycle until the termination criterion is satisfied.

In step 2.1, PC individual exploration is addressed and two crucial questions are considered. One is how to select promising Pareto non-dominated individuals from the PC population, and the other is how to produce the offspring. For the first question, those PC individuals whose niche has no or only one NPC individual are selected in order to probe into the regions that are not well developed in the NPC evolution. For the second question, problem-specific crossover operators and neighborhood generation operators described in section 4.4 are designated to achieve global evolutionary search. In step 2.2, the NPC population evolution is implemented by using MOEA/D (Li & Zhang, 2009) with embedded problem-specific variation operator, which are the same as those in step 2.1. In step 2.3, a niche-based PC population maintenance strategy is proposed to preserve a set of representative PC individuals, which maximizes the diversity of the PC population and maintains a fixed PC population size.

A clear difference between the proposed PBEA and BCE is that PBEA is a specific algorithm, whereas BCE is a general framework. In addition, PBEA differs from BCE by a significant margin in exploration in the decision space. BCE focuses on the implementation of environmental selection operations, whereas the variation operation in the decision space have not been explored. It would be essential to utilize problem-specific information to guide the search process of general-purpose BCE so as to enhance the exploration ability. In PBEA, problem-specific crossover operators, neighborhood generation operators and repair procedures are developed for MDLB-MR through well-designed solution encoding and decoding, enhancing the exploration ability of the decision space.

Algorithm 1 Framework of the proposed PBEA for MDLB-MR.

Input: TAOG models; task processing times; energy consumptions of robots; setting of MDL-MR.

Output: Approximation to Pareto set and Pareto front.

- 1: **Step 1 Initialization** initialize NPC population npc_pop by problem-specific method (Section 4.3), initialize reference point and PC population pc_pop .
- 2: **Step 2 Update**
- 3: Step 2.1 PC population evolution
- 4: Step 2.1.1 Calculate the radius of the niche for each PC individual.
- 5: Step 2.1.2
- 6: **for** (all $x_i \in pc_pop$) **do**
- 7: **if** (no or only one NPC individual in the niche of x_i) **then**
- 8: Randomly select x_j from pc_pop , then $x_k \leftarrow variation(x_i, x_j)$ (Section 4.4) and update pc_pop , reference point, npc_pop by child x_k .
- 9: **end for**
- 10: Step 2.2 NPC population evolution
- 11: **for** ($i = 1 \dots |npc_pop|$) **do**

- 12: Step 2.2.1 **if** ($\text{randreal}(0, 1) < \text{threshold}$) **then** $\text{candidateSet} \leftarrow \text{neighbors of } i$ **else** $\text{candidateSet} \leftarrow \{1 \dots |\text{npc_pop}|\}$.
 - 13: Step 2.2.2 Randomly select **an index** j from candidateSet , then $x_k \leftarrow \text{variation}(x_i, x_j)$.
 - 14: Step 2.2.3 Update pc_pop and **reference point** by **child** x_k , update npc_pop based on MODE/D .
 - 15: **end for**
 - 16: Step 2.3 PC population maintenance (Shown in **BCE framework**)
 - 17: **Step 3 Stopping Criterion** **if** (the stopping criterion is satisfied) **then** **stop and output** **else** go to Step 2.
-

4.2. Solution encoding and decoding

To obtain a feasible **solution**, the assignment of tasks to robots, the processing sequence of tasks on robots, and the allocation of the robots to workstations should be determined. **In this study, three integer vectors is developed for solution encoding: task-robot assignment vector, task sequence vector and robot-workstation assignment vector.**

Before explaining the three vectors, the task unified numbering for **all EOL product models** should be addressed. **In MDLB-MR**, a set of models M can be simultaneously completely disassembled. We consecutively give a fixed global ID (GID) for **each task** in TAOGs of all models in the form b , where $b \in B$ with $|B| = \sum_{m \in M} |B^m|$. This means that tasks $1 \dots |B^1|$ belong to model 1, tasks $|B^1| + 1 \dots |B^1| + |B^2|$ belong to model 2 and so on. After unified numbered, a task can also be referred to the fixed GID from 1 to $|B|$. For example, in Fig. 3, task GID 3 ($b = 3$) has the same reference with task B_3^1 in TAOG of **model 1**.

Task-robot assignment vector and task sequence vector together determine the assignment of tasks on the robot. The task-robot assignment vector, which is denoted by $TR = [r_1 \dots r_d \dots r_{|D|}]$, is an array of $|D|$ integer values, where $r_d \in R$ and $d = 1 \dots |D|$. Task assignment vector can determine the assignment of the tasks to robots, but precedence relations of tasks on robots remain unsolved and the task sequence vector is thus utilized. The task sequence vector $TS = [b_1 \dots b_d \dots b_{|D|}]$ is a GID permutation of $|D|$ selected tasks, where $b_d \in B$ and $d = 1 \dots |D|$. The order of occurrence for each task in TS indicates its processing priority. For example, in the task sequence vector TS shown in Fig. 3, task GID 3 has the highest priority and it is assigned first, then task GID 24, and so on. Note that the selected tasks within a TAOG should keep the relative precedence relations in the TS , and they also should be able to form a **DT**. Moreover, there is a one-to-one correspondence between the elements of the same index in TR and TS . That is to say, if the d^{th} element r_d in TR is r and the d^{th} element b_d in TS is b , then **task** b is supposed to be assigned to robot r . In order to determine the allocation of the robots to workstations, the robot-workstation assignment vector $RW = [j_1 \dots j_r \dots j_{|R|}]$, an array of $|R|$ integer values, where

$j_r \in J \cup \{-1\}$ and $r \in R$, is proposed. Noted that the r^{th} element j_r in RW is '-1' indicates robot r is not allocated to any workstation. In other words, robot r is an idle resource.

Fig. 3. Example of solution encoding and decoding for MDLB-MR ($|M| = 2$, $|B| = 43$, $|D| = 15$, $|J| = 2$, $|R| = 10$ and $R_{max} = 3$).

Furthermore, a decoding scheme is proposed to allocate a period of time for each selected task on its assigned robot one by one according to their occurrence order in TS . For task b and its assigned robot r , the basic idea behind the proposed decoding scheme is scanning time dimension of robot r from time 'zero' and obtaining the first available idle time interval between tasks that have already been allocated on robot r for task b (Yuan & Xu, 2015). Note that robots in the same workstation have the same time 'zero', while the time 'zero' of robots in one workstation differs by a cycle time from that in the adjacent workstation. This means that the working time interval, which includes operation time and standby time, of robots in workstation 1 is $[0, CT]$, the working time interval of robots in workstation 2 is $[CT, 2 \cdot CT]$ and so on. For any selected task b in TS ($b = b_d$) and its assigned robot r ($r = r_d$), its available idle time interval $[L_t, R_t]$ is determined by using the formula (1):

$$\begin{cases} \max\{L_t, C_{P(b, TS)}\} + t_{br} \leq R_t, & \text{if } \exists P(b, TS); \\ L_t + t_{br} \leq R_t, & \text{otherwise.} \end{cases} \quad (1)$$

where L_t and R_t represent the starting and ending time of the interval respectively, t_{br} is the processing time of task b by robot r and $C_{P(b, TS)}$ is the completion time of the immediate normal predecessor of task b in TS . Actually, in any selected DT, there is no immediate task predecessor for root task and only one immediate task predecessor for any other task. Hence, at most, each selected task has one immediate predecessor in TS .

Once a selected task b in TS is to be allocated in the first available idle time interval on its assigned

robot r , its starting time will be set. If there is no such idle time interval, task b would be allocated at the end of all tasks that have already been allocated to robot r . It can be easily proved that our decoding scheme can not only guarantee that the allocation of tasks to robot is an active **schedule**, but also be beneficial to the objectives $F1$, $F2$ and $F3$ of MDLB-MR simultaneously. Note that the objective $F4$ of MDLB-MR is fixed for any given solution. Therefore, our decoding scheme is feasible. When a solution is decoded, the tasks in **TS** are reordered in an ascending order of their starting time.

An example of solution representation for MDLB-MR is depicted in Fig. 3. In this example, 10 robots can be allocated to 2 workstations. From RW in Fig. 3, we can see that some robots, robot 2, 3, and 4, are allocated to workstation 1, some are allocated to workstation 2 such as robot 5 and 6, and **the others** are idle. Furthermore, **15 selected tasks from** TAOGs of 2 models as shown in Fig. A2 can also be assigned exactly to robots based on **TS/TR** in Fig. 3 and the data provided by Appendix B.

4.3. Solution initialization

The solution initialization procedure begins with randomly generating a feasible RW . First, the number of robots ($|R_j|$) allowed in **workstation j** is randomly generated from the integer interval $[1, R_{max}]$. Then, $|J|$ iterations are performed to achieve the allocation of robots to the given $|J|$ workstations. In each iteration j , $|R_j|$ robots are selected randomly from the robot candidate list and then the candidate list is updated by deleting the selected robots. Note that the robot candidate list contains all $|R|$ robots at the beginning of the iterations and $|R|$ should always be greater than or equal to $|J| \cdot R_{max}$. If robot r is not allocated to any workstation, the r^{th} element j_r in RW will be set to -1 .

Fig. 4. Example of the DT generation procedure.

To obtain feasible TS and TR , $|M|$ DTs need to be generated first and each of them is selected randomly from the corresponding TAOG of **a model**. The DT generation procedure is the same for all $|M|$ models and starts with randomly selecting **a task** from the successors of the first artificial node in TAOG. Then the immediate successor set of the **selected task** is determined and the number of artificial nodes in the set **is no more than two**. For each artificial node in the set, **a new task** is selected randomly from its successors. This procedure is repeated until $|P^m| - 1$ **tasks are** selected and the output represents

a DT. Fig. 4 illustrates the DT generation procedure on a sample TAOG in Fig. A2(a). As seen in Fig. 4 tasks 3, 9, 18, 19, 22 and 23 are randomly chosen to compose a DT.

Based on these $|M|$ DTs and the RW , the following task-robot assignment procedure, originating from the idea in Nilakantan et al. (2015) and Roshani & Giglio (2017), is proposed to generate feasible TR and TS heuristically.

- (1) Calculate the initial cycle time by using formula (2). **Note that**, CT_0 should always be greater than or equal to the mean of the minimum processing time of robots for all $|D|$ selected tasks in $|M|$ DTs and the maximum processing time among these selected tasks. The initial processing time of task is set to the minimum processing time of robots for it.

$$CT_0 = \max\left\{ \max_{b=b_d \in B, d=1 \dots |D|} \left\{ \min_{r \in R, j_r \in J} \{t_{br}\} \right\}, \left\lceil \frac{\sum_{b=b_d \in B, d=1 \dots |D|} \min_{r \in R, j_r \in J} \{t_{br}\}}{|J| \cdot R_{max}} \right\rceil \right\} \quad (2)$$

- (2) Set the current index j of workstation as zero. Determine **the set of unassigned tasks** ($setB_{DT} \leftarrow D$).
- (3) If $j = |J|$ and $setB_{DT} = \emptyset$, then terminate the procedure. Generate TS by sorting all $|D|$ selected tasks in $|M|$ DTs in a non-decreasing order of task starting time (if tie occurs, then sort these tasks with the same starting time according to the **ranked positional weight**). Generate TR according to the result of the task-robot assignment. Else:
 - (3.1) If $j \leq |J| - 1$, then set $j = j + 1$;
 - (3.2) If $j = |J|$ and $setB_{DT} \neq \emptyset$, then set $CT_0 = CT_0 + \text{time increment}$. Go to (2).
- (4) Determine **the set of available robot indices** ($setR_j$) allocated to workstation j from RW . Set $C_r = 0$ for any $r \in setR_j$, where C_r denotes **the task completion time of robot r** .
- (5) Determine **the set of assignable tasks** ($setB_{DTA}$) which includes those tasks in set $setB_{DT}$ having no predecessor, or their predecessors have been assigned before.
- (6) Calculate the completion time of each task in $setB_{DTA}$ for all $r \in setR_j$ by using formula (1). Exclude the tasks that violate the cycle time constraint for all $r \in setR_j$ in workstation j from $setB_{DTA}$.
- (7) If $setB_{DTA} = \emptyset$, then go to (3); else, select the task b with the highest ranked positional weight from $setB_{DTA}$. In this study, the ranked positional weight of a task is the sum of the processing times of the task and of all its successors in $setB_{DT}$. The processing time of task is the mean of the processing time of all robots from available robot set for this task.
- (8) According to the formula (1), assign the selected task b to the robot $r \in setR_j$ that can complete it earlier (if tie occurs, then select a robot with lower energy consumption). Update C_r and remove b^{th} task from $setB_{DT}$. Go to (5).

4.4. Problem-specific variation operators

The variation operators in PBEA consist of the crossover and neighborhood generation operators. The crossover operators are applied for RW and TS to emphasize exploration in the decision space, while the neighborhood generation operators, including alteration and insert operators, are applied for all the three vectors to provide a neighborhood search on the current solution.

The crossover operators for RW and TS are implemented respectively. For RW , a subset of indices is selected from one parent at random. Then a child is produced by copying the values of these selected indices from the parent to the corresponding indices in the child. Finally, the values of the remaining indices are copied from the second parent to the corresponding indices in the child. Noted that the obtained RW may not be feasible due to the constraints on the maximum number of robots in each workstation and the fixed number of workstations. So, a repair procedure is introduced to adjust robot-workstation assignment of the obtained RW . The repair procedure has the following major steps:

- (1) Set the current index j of workstation as zero.
- (2) If $j = |J|$, then terminate the procedure and output the modified RW ; else set $j = j + 1$.
- (3) Determine the robot set $setR_j$ allocated to station j from RW . If the size of $setR_j$ is greater than R_{max} , then randomly select $(|setR_j| - R_{max})$ robots from $setR_j$ and deallocate them from workstation j . If the size of $setR_j$ equals zero, then randomly select $randint(1, R_{max})$ robots from the set of all unallocated robots to allocate to workstation j . Go to (2).

Fig. 5. Illustration of the crossovers for RW and TS ($|M| = 2$, $|B| = 43$, $|D| = 15$, $|J| = 2$, $|R| = 10$ and $R_{max} = 3$).

With regards to TS , the crossover operator allows each DT in a child to be selected from one parent according to predetermined probability or from the second parent in other cases. Once all new $|M|$ DTs are determined, the task-robot assignment procedure, which is described in section 4.3, will be triggered

to generate new suboptimal TS and TR . The above crossover operators for RW and TS are illustrated in Fig. 5. In this study, both these two crossover operators would be simultaneously triggered with a given probability for each iteration to not only explore new patterns of a solution to enhance the search ability but also inherit parents' features as much as possible.

Unlike above crossover operators, neighborhood generation operators are applied for all the three vectors by modifying values of different positions within a solution. A swap operator is applied for RW to perform interchange between the values of the workstation that an assigned robot belongs to and an unassigned robot in RW . For TR , an alteration operator is applied. In the alteration operator, a task is randomly selected and the robot of this task is replaced with another assigned robot which is randomly selected from the available robot set of the task, by checking the robots and workstations of its immediate predecessors and successors. As for TS , a neighbor solution is obtained by an insert operator. The insert operator selects a task in TS randomly and moves it from the current position to a different feasible position which cannot violate the precedence relationships between this task and its immediate predecessor and successor in TS .

In addition, a simple method (Li et al., 2016b) is applied to judge whether a neighborhood generation operator can be selected for each iteration. A number between $[0, 1]$ is generated randomly for each vector. If the number is smaller than $1/3$, then the neighborhood generation operator for the corresponding vector is selected and triggered. If all the three numbers are larger than $1/3$, the operator for the RW is triggered.

5. Computational experiments

5.1. Experimental settings

First of all, the generation of test instances needed to be considered. To our best knowledge, MDLB-MR is firstly explored in this paper. There are no standard test datasets in this area. In this study, 8 models of realistic EOL products from the literature, namely, model 1 (Koc et al., 2009), model 2 (Lambert, 1997), model 3 (Tang et al., 2002), model 4 (Lambert, 2007), model 5 (Lambert, 1999), model 6 (Lambert, 1999), model 7 (Ma et al., 2011) and model 8 (Lambert, 2007) are used to illustrate the effectiveness of PBEA. A new test dataset including 63 problem instances for MDLB-MR is generated by using 8 TAOGs of these products. The dataset is divided into three groups: 19 small-size instances with 2 – 3 EOL product models (23 – 73 possible tasks), 24 medium-size instances with 4 – 5 models (65 – 132 tasks) and 20 large-size instances with 6 – 8 models (118 – 187 tasks). To show the superiority of TAOG over classical AND TPD in MDLB-MR, the AND TPDs of top 4 models and the task processing times by robots are also generated by using the AND TPD derivation method developed by Koc et al. (2009). The proposed instances are processed on a disassembly line with 4 multi-robotic workstations.

There are 25 available robots and it is possible to assign up to 5 different robots to each workstation. The processing times of tasks by robots are randomly generated between $[t_{bro} \cdot 0.7, t_{bro} \cdot 1.3]$ and t_{bro} is the referenced processing time published. The operation energy consumptions of robots per time unit are shown in Appendix B and the standby energy consumption of a robot per time unit can be considered to 10% of the operation energy consumption (Nilakantan et al., 2017).

In the field of evolutionary multi-objective optimization, the selection criterion plays a key role and there are two classes employed to select individuals in the population: Pareto and non-Pareto criteria. Additionally, the indicator-based and decomposition-based EAs are two representative classes in using the non-Pareto criteria to deal with multi-objective optimization. In view of this, in this paper, three representative EAs, NSGA-II (Deb et al., 2002), IBEA (Zitzler & Künzli, 2004) and MOEA/D (Li & Zhang, 2009), which lead the evolution via the Pareto criterion, indicator-based criterion, and decomposition-based criterion respectively, are introduced and re-implemented for the algorithm comparative study. Furthermore, in MOEA/D, three scalarizing functions including weighted sum (WS), penalty-based boundary intersection (PBI) and Tchebycheff (TCH) functions are considered in our experiments to investigate the implications of different scalarizing functions on algorithm performance. Note that the comparative study focuses on the different selection operations, whereas the other components such as encoding/decoding scheme, solution initialization approach and variation operators in these competitors are the same and stem from PBEA. All the algorithms are implemented in C programming language by Xcode, and tested on 8 Intel Core i7 3.1 GHz processor with 16 GB of RAM. All of the above data and algorithm codes are available upon request to the authors.

HV is used to evaluate the performance of these algorithms in this study. It is a very popular metric in the evolutionary multi-objective optimization area due to its good theoretical and practical properties, such as being compliant with Pareto dominance (if one population Pareto dominates another then it has a higher HV) and not requiring the problem's Pareto front to be known. HV calculates the volume of the objective space between the obtained solution set and a specified reference point. A large HV value is preferable and reflects good performance of the solution set in terms of convergence, extensity and uniformity. Note that the exact Pareto fronts of these many-objective problem instances are unknown and so for the calculation of HV we use a collection of all the solutions obtained by executing all the runs for all the algorithms and pick the non-dominated solutions from the collection as the ideal solution set to obtain the approximate information about the Pareto front. Then we normalize the objective value of the obtained solutions according to the range of the approximate Pareto front and calculate HV value based on the normalized value. Also, the reference point is set to 1.1 times of the nadir point of the obtained Pareto front. To test the difference for statistical significance in all the considered 63 test instances, the

Wilcoxon's rank sum test at a 5% significance level is conducted on the HV values obtained by PBEA and each competitor.

All the results presented in the experimental study are obtained by executing 10 independent runs for each algorithm. The termination criterion used in each algorithm is a predefined number of function evaluations, which is set to 100,000. For a fair comparison, the size of the population for IBEA and NSGA-II is set to 100. Since the setting of population size for MOEA/D and PBEA cannot be arbitrarily specified, where it is controlled by division number of each objective axis, we use the closest in integer to 100 amongst the possible values as the population size of the 3 versions of MOEA/D and PBEA. Besides the parameters mentioned above, there are other parameters need to be set. These parameters are set mainly according to the suggestions given by developers, which are shown in Table 7.

Table 7

Other parameter of the considered algorithms.

Parameter	Algorithms	Value	Parameter	Algorithms	Value
Scaling factor	IBEA	0.05	Niche radius	PBEA	3
Neighborhood size	MOEA/D,PBEA*	12	Increment of cycle time	ALL [†]	4.0
Probability of neighborhood selection	MOEA/D,PBEA*	0.9	RW generation probability in crossover	ALL [†]	0.6
Number of divisions	MOEA/D,PBEA*	7	DT selection probability in crossover	ALL [†]	0.6
Maximum number of replacement	MOEA/D,PBEA*	1	Mutation operators selection probability	ALL [†]	0.33
Penalty parameter	MOEA/D+PBI	5	Probability of TS search direction selection	ALL [†]	0.5

"*" means that the algorithms include the 3 versions of MOEA/D and PBEA; "†" indicates that the algorithms include all 6 algorithms.

5.2. Experimental results and discussions

5.2.1. Investigation of TAOG and AND TPD on MDLB-MR

To investigate the impact of TAOG and AND TPD approaches in the representation of task precedence relations on PBEA, we consider 4 MDLB-MR problem instances including from 2 to 4 EOL product models. For each instance, the approximate Pareto front is obtained firstly. Then the normalized non-dominated solutions over 10 runs of PBEA based on TAOG and AND TPD are calculated, respectively. Fig. 6 plots the final experimental results of the TAOG and AND TPD approaches on these 4 problem instances by parallel coordinates. Parallel coordinates have been found to be a useful tool for visualizing many-objective solutions in a two-dimensional graph. Each line in the graph connects the objective values achieved by an individual and represents a potential solution to the given problem. In Fig. 6 where the order of objectives on the horizontal axis is $F1$, $F2$, $F3$, $F4$, the vertical axis represents the normalized

objective value. Note that “ m, m' ” in each subtitle of Fig.6 denotes model m and model m' .

Fig. 6. Normalized non-dominated solution set obtained by PBEA based on TAOGs or AND TPDs on different mixed-model scenarios (2-4 models), as shown by parallel coordinates.

From Fig. 6(a)-6(h), we can see the PBEA based on TAOG shows a clear advantage over the PBEA based on AND TPD on all 4 problem instances, especially on the first three objectives. It is straightforward to see the PBEA based on AND TPD has an inferior convergence, with its final non-dominated solution set for each problem instance ranging from 0 to around 3.5, 4.0 or 4.5, in contrast to the Pareto front ranging from 0 to 1, whereas the PBEA based on TAOG has a good convergence and its solution set for each instance has the same range as the Pareto front. Moreover, all the solution sets obtained by the PBEA based on TAOG for all 4 problem instances have good coverage and they all spread over the whole range of the desired front. In contrast, for the PBEA based on AND TPD, all the solution sets fail to cover the region from 0 to around 0.5 on objective $F1$ and $F3$. Especially, there are no points distributed between 0 and 1 on objective $F2$. As the illustrative example in section 2.3, the experimental results in this section also show the remarkable superiority of using TAOG instead of AND TPD to model the task precedence relations. This is consistent with the optimal result in Fig. 2 and it is mainly because that all three properties listed in section 2.3, especially parallel disassembly described in the third property, can be utilized to improve the performance of MDLB-MR.

5.2.2. Comparison with state-of-the-art evolutionary algorithms

In this section, we observe the behavior of different EAs on MDLB-MR by comparing PBEA with five representative EAs, i.e., IBEA, NSGA-II, MOEA/D+WS, MOEA/D+PBI and MOEA/D+TCH. Tables 8, 9 and 10 give the HV results (mean and standard deviation), for all the 6 algorithms on the three groups

of instances. Note that the first column of the tables gives the EOL product models, where “ m, m' ” denotes model m and model m' . Moreover, the better mean for each problem instance is highlighted in boldface. In the three tables, the results significantly differ from those of PBEA are marked in symbol “†”.

Table 8

HV results (mean [standard deviation]) of the 6 algorithms for small sized instances.

Models	IBEA	NSGA-II	MOEA/D+WS	MOEA/D+PBI	MOEA/D+TCH	PBEA
1,2	8.6703e-01[6.4e-02]	9.5698e-01[4.1e-02] †	6.0495e-01[8.0e-02]†	4.4322e-01[4.9e-02]†	7.3045e-01[1.2e-01]†	8.9281e-01[5.4e-02]
1,3	9.1447e-01[8.6e-02]	9.5826e-01[7.3e-02]	6.7410e-01[1.1e-01]†	5.3005e-01[1.1e-01]†	8.6672e-01[6.6e-02]†	9.3389e-01[1.0e-01]
1,4	9.1300e-01[1.5e-02]†	9.5206e-01[1.8e-02] †	6.4240e-01[9.5e-02]†	4.3921e-01[9.8e-02]†	7.6921e-01[1.1e-01]†	9.1942e-01[5.2e-02]
2,3	8.6493e-01[2.1e-02]†	9.1038e-01[1.5e-02]	7.0129e-01[6.3e-02]†	5.5792e-01[8.9e-02]†	6.8495e-01[4.5e-02]†	9.0545e-01[2.3e-02]
2,4	1.0499e+00[2.6e-02]	1.0684e+00[1.6e-02] †	6.3167e-01[1.4e-01]†	4.6508e-01[5.6e-02]†	9.3071e-01[1.2e-01]†	1.0537e+00[1.3e-02]
3,4	9.0878e-01[4.9e-02]†	9.6711e-01[1.3e-02]	7.4330e-01[6.3e-02]†	5.1304e-01[1.9e-01]†	7.7562e-01[1.4e-01]†	9.5690e-01[2.6e-02]
2,5	8.5431e-01[6.1e-02]	9.0600e-01[3.4e-02]	6.2200e-01[1.1e-01]†	4.2192e-01[8.6e-02]†	6.1930e-01[8.5e-02]†	8.6825e-01[4.4e-02]
1,2,3	8.7069e-01[3.9e-02]	8.7583e-01[3.2e-02]	5.9709e-01[9.8e-02]†	3.6364e-01[8.6e-02]†	5.8520e-01[9.1e-02]†	8.5630e-01[5.3e-02]
1,2,4	8.2705e-01[3.6e-02]†	8.2390e-01[8.8e-02]	5.3481e-01[6.8e-02]†	3.5242e-01[5.6e-02]†	5.7751e-01[9.6e-02]†	8.7800e-01[3.0e-02]
1,2,5	9.0460e-01[6.4e-02]	9.3077e-01[4.0e-02]	6.7747e-01[9.2e-02]†	5.5309e-01[9.6e-02]†	7.4869e-01[8.9e-02]†	9.3434e-01[4.2e-02]
1,2,6	9.2677e-01[4.2e-02]†	9.5082e-01[3.6e-02]	7.2203e-01[9.8e-02]†	5.9353e-01[5.8e-02]†	6.9767e-01[1.0e-01]†	9.6394e-01[3.3e-02]
1,3,4	9.6117e-01[3.5e-02]†	1.0136e+00[4.6e-02]	8.3875e-01[7.3e-02]†	6.8412e-01[7.2e-02]†	8.2622e-01[9.0e-02]†	1.0114e+00[4.4e-02]
1,3,6	8.3100e-01[3.9e-02]†	8.5865e-01[6.1e-02]	6.9274e-01[1.0e-01]†	5.2255e-01[1.2e-01]†	6.5424e-01[1.1e-01]†	8.7769e-01[4.0e-02]
1,4,6	8.2529e-01[5.4e-02]	8.7695e-01[3.0e-02]	6.7530e-01[8.6e-02]†	5.0815e-01[7.8e-02]†	6.5895e-01[9.2e-02]†	8.5134e-01[5.5e-02]
2,3,4	9.0753e-01[4.0e-02]	9.5987e-01[5.0e-02]	7.1546e-01[6.5e-02]†	4.7944e-01[1.2e-01]†	6.9842e-01[6.1e-02]†	9.3198e-01[5.7e-02]
2,3,5	9.2576e-01[3.5e-02]†	9.4893e-01[4.0e-02]	7.6546e-01[5.2e-02]†	5.5364e-01[9.0e-02]†	8.2426e-01[6.9e-02]†	9.7205e-01[4.2e-02]
2,3,6	7.9607e-01[4.3e-02]†	8.4815e-01[4.4e-02]	6.2104e-01[1.2e-01]†	4.8912e-01[8.1e-02]†	5.9131e-01[1.0e-01]†	8.5137e-01[4.7e-02]
2,4,5	7.9744e-01[8.3e-02]	8.5629e-01[6.0e-02]	6.7357e-01[1.0e-01]†	4.2664e-01[8.1e-02]†	6.2019e-01[1.2e-01]†	8.4034e-01[5.4e-02]
2,4,6	9.2253e-01[3.9e-02]	9.3386e-01[5.2e-02]	7.3832e-01[6.2e-02]†	5.5656e-01[7.4e-02]†	7.4041e-01[8.0e-02]†	9.6125e-01[4.1e-02]

“†” means that the result is significantly different from that of the PBEA at a 0.05 level by the Wilcoxon's rank sum test.

The HV results of the 6 algorithms for all 19 small sized instances are shown in Table 8. From the Table 8, it is observed that, NSGA-II and PBEA are the best two algorithms. Specifically, NSGA-II is overall slightly better than PBEA, where the NSGA-II wins in 12 out of 19 small sized instances, while the PBEA outperforms the others in the rest 7 small sized instances. Tables 9 and 10 give the HV results of all algorithms for 24 medium sized instances and 20 large sized instances, respectively. As can be seen in Tables 9 and 10, PBEA generally performs better than its competitors on the majority of the medium-size and large-size instances. Specifically, PBEA obtains a better HV value in 15 out of the 24 medium sized instances and 17 out of the 20 large sized instances, respectively. All these results indicate the superiority of PBEA regarding the scalability and it can be thought that PBEA can be applied to medium or large scale MDLB-MR problems successfully. In addition, NSGA-II shows the closest overall performance to PBEA, and it can also perform well over a wide range of test problem instances. There is no significant difference between the HV results of NSGA-II and PBEA in most of the instances. Then IBEA has the medium-high performance on most of the instances among the compared algorithms. It is worth mentioning that all these three algorithms significantly outperform the three MOEA/D variants,

i.e., MOEA/D+WS, MOEA/D+PBI and MOEA/D+TCH. In fact, the three MOEA/D variants cannot produce satisfactory results on all of the tested instances and Tables 8, 9 and 10 also show that they are the worst three ones among considered EAs. Furthermore, it can be observed that MOEA/D+TCH achieves a better HV value than MOEA/D+WS and MOEA/D+PBI on most of the instances.

Table 9

HV results (mean [standard deviation]) of the 6 algorithms for medium sized instances.

Models	IBEA	NSGA-II	MOEA/D+WS	MOEA/D+PBI	MOEA/D+TCH	PBEA
1,2,3,4	8.7690e-01[3.2e-02]	8.8305e-01[2.3e-02] [†]	6.9757e-01[8.5e-02] [†]	4.9346e-01[9.6e-02] [†]	6.7996e-01[1.1e-01] [†]	9.1211e-01[3.2e-02]
1,2,3,5	9.4153e-01[3.2e-02] [†]	9.6455e-01[4.6e-02]	7.4701e-01[4.0e-02] [†]	5.8260e-01[9.0e-02] [†]	7.6434e-01[8.5e-02] [†]	9.7309e-01[1.7e-02]
1,2,3,6	8.8947e-01[6.5e-02]	8.7403e-01[9.4e-02]	6.1753e-01[6.6e-02] [†]	4.5401e-01[7.4e-02] [†]	6.6553e-01[8.9e-02] [†]	8.8270e-01[6.0e-02]
1,2,4,5	9.6256e-01[4.1e-02] [†]	9.8948e-01[2.9e-02]	7.9872e-01[7.0e-02] [†]	6.3326e-01[5.5e-02] [†]	7.8769e-01[6.4e-02] [†]	1.0072e+00[4.6e-02]
1,2,4,6	8.9317e-01[5.0e-02]	9.2691e-01[5.4e-02]	6.7114e-01[9.8e-02] [†]	6.0755e-01[9.9e-02] [†]	7.0376e-01[7.3e-02] [†]	9.1373e-01[5.4e-02]
1,4,5,6	7.0295e-01[7.7e-02]	6.8957e-01[8.7e-02]	4.4668e-01[1.6e-01] [†]	3.5178e-01[1.4e-01] [†]	4.4228e-01[1.5e-01] [†]	6.9361e-01[9.3e-02]
2,3,4,5	8.7180e-01[3.5e-02] [†]	9.1544e-01[3.3e-02]	7.3558e-01[6.3e-02] [†]	5.4973e-01[8.0e-02] [†]	6.7683e-01[4.8e-02] [†]	9.1799e-01[3.6e-02]
2,3,4,6	9.5067e-01[4.6e-02]	9.7634e-01[3.8e-02]	7.2910e-01[9.8e-02] [†]	5.5765e-01[8.0e-02] [†]	7.9895e-01[9.6e-02] [†]	9.7660e-01[6.3e-02]
3,4,5,6	8.3033e-01[5.2e-02] [†]	9.0101e-01[6.3e-02]	6.3819e-01[8.3e-02] [†]	3.9844e-01[1.0e-01] [†]	6.1932e-01[8.7e-02] [†]	8.9630e-01[4.7e-02]
1,2,3,4,5	8.4871e-01[6.5e-02]	9.0899e-01[3.9e-02]	6.8293e-01[6.2e-02] [†]	5.2779e-01[6.4e-02] [†]	7.0127e-01[7.9e-02] [†]	8.8962e-01[4.0e-02]
1,2,3,4,6	8.5129e-01[6.6e-02]	8.9627e-01[5.6e-02]	6.6253e-01[7.9e-02] [†]	5.2485e-01[8.7e-02] [†]	6.9045e-01[7.5e-02] [†]	8.9623e-01[4.8e-02]
1,2,3,4,7	9.2086e-01[3.0e-02] [†]	9.5924e-01[4.0e-02]	7.2912e-01[4.9e-02] [†]	5.3231e-01[9.3e-02] [†]	7.0746e-01[1.0e-01] [†]	9.5850e-01[3.0e-02]
1,2,3,4,8	9.5743e-01[3.8e-02] [†]	9.8918e-01[2.6e-02]	7.4944e-01[7.3e-02] [†]	5.8556e-01[6.1e-02] [†]	8.0625e-01[8.1e-02] [†]	1.0160e+00[5.0e-02]
1,2,3,5,6	8.4762e-01[5.0e-02] [†]	8.7009e-01[5.0e-02]	6.3567e-01[6.8e-02] [†]	6.1187e-01[8.8e-02] [†]	7.0777e-01[8.7e-02] [†]	9.1651e-01[5.9e-02]
1,2,3,5,7	8.4560e-01[4.4e-02]	8.6655e-01[5.8e-02]	6.4482e-01[8.8e-02] [†]	4.8562e-01[7.0e-02] [†]	6.3575e-01[8.9e-02] [†]	8.7384e-01[6.6e-02]
1,2,3,5,8	8.8673e-01[3.2e-02] [†]	9.2922e-01[4.7e-02]	6.9142e-01[6.5e-02] [†]	5.0630e-01[1.1e-01] [†]	6.7061e-01[9.7e-02] [†]	9.5586e-01[3.4e-02]
1,2,4,5,6	9.5790e-01[5.9e-02]	9.5495e-01[4.1e-02]	7.2222e-01[7.5e-02] [†]	5.3309e-01[7.7e-02] [†]	7.4241e-01[6.2e-02] [†]	9.8707e-01[3.9e-02]
1,2,4,5,7	9.0355e-01[2.9e-02]	9.1818e-01[3.9e-02]	6.7306e-01[8.2e-02] [†]	5.3392e-01[6.7e-02] [†]	6.8354e-01[7.0e-02] [†]	9.3054e-01[3.9e-02]
1,2,4,5,8	8.6093e-01[8.2e-02]	8.8292e-01[6.4e-02]	6.0137e-01[8.2e-02] [†]	4.5811e-01[7.3e-02] [†]	6.6058e-01[9.3e-02] [†]	9.0876e-01[5.4e-02]
1,2,5,6,7	8.4908e-01[4.3e-02]	8.8339e-01[7.4e-02]	6.1214e-01[6.7e-02] [†]	5.3596e-01[7.2e-02] [†]	6.4053e-01[7.0e-02] [†]	8.8554e-01[5.0e-02]
1,2,5,6,8	8.1305e-01[6.5e-02]	8.7069e-01[2.0e-02]	5.8941e-01[8.7e-02] [†]	4.5738e-01[1.0e-01] [†]	5.9408e-01[1.2e-01] [†]	8.1943e-01[7.6e-02]
2,3,4,5,6	8.8921e-01[6.6e-02]	9.0066e-01[6.8e-02]	7.2808e-01[6.1e-02] [†]	5.7848e-01[6.2e-02] [†]	7.2931e-01[8.8e-02] [†]	9.1163e-01[4.2e-02]
2,3,4,5,7	9.3046e-01[3.3e-02]	9.3374e-01[3.4e-02]	7.3311e-01[6.2e-02] [†]	5.6306e-01[5.8e-02] [†]	7.5214e-01[8.5e-02] [†]	9.7440e-01[5.6e-02]
2,3,4,5,8	8.0531e-01[8.8e-02]	8.8181e-01[3.3e-02]	6.1204e-01[1.0e-01] [†]	4.5737e-01[1.2e-01] [†]	5.7986e-01[1.1e-01] [†]	8.6489e-01[7.2e-02]

“†” means that the result is significantly different from that of the PBEA at a 0.05 level by the Wilcoxon's rank sum test.

To describe the distribution of solutions in high-dimensional objective space more intuitively, we use MDLB-MR instance “1, 2, 3, 4, 5, 6, 7, 8” as an illustration. Fig. 7 plots the final solutions of all these EAs in a single run by parallel coordinates. This particular run is associated with the result closest to the average HV value. It is clear from Fig. 7 that both PBEA and NSGA-II are able to find a good approximation and coverage of the Pareto front represented by $F_i \in [0, 1]$ for all four objectives ($i = 1 \dots 4$). Furthermore, a better tradeoff among the solutions obtained by PBEA is clear from Fig. 7(f). In contrast, IBEA and the three MOEA/D variants can only converge to a portion of the Pareto front.

Based on the above observations, some insights with respect to the design of EAs for MDLB-MR have been gained, including the followings:

Table 10

HV results (mean [standard deviation]) of the 6 algorithms for large sized instances.

Models	IBEA	NSGA-II	MOEA/D+WS	MOEA/D+PBI	MOEA/D+TCH	PBEA
1,2,3,4,5,6	8.5405e-01[4.5e-02]	8.5461e-01[5.6e-02]	6.2393e-01[6.6e-02] [†]	4.9336e-01[8.6e-02] [†]	6.7624e-01[1.0e-01] [†]	8.8997e-01[5.9e-02]
1,2,3,4,5,7	8.9422e-01[5.3e-02]	8.6959e-01[4.4e-02]	6.7473e-01[7.8e-02] [†]	5.3028e-01[8.4e-02] [†]	6.9436e-01[7.2e-02] [†]	8.9671e-01[4.1e-02]
1,2,3,4,5,8	8.8235e-01[6.3e-02]	9.1127e-01[7.2e-02]	6.2041e-01[8.9e-02] [†]	5.0715e-01[8.5e-02] [†]	7.0166e-01[1.0e-01] [†]	9.3805e-01[4.6e-02]
1,2,3,4,6,7	8.8978e-01[2.7e-02] [†]	9.0159e-01[6.4e-02]	7.2553e-01[4.5e-02] [†]	4.5334e-01[5.1e-02] [†]	6.9378e-01[6.7e-02] [†]	9.2926e-01[3.4e-02]
1,2,3,4,6,8	8.6238e-01[7.5e-02]	8.3686e-01[5.3e-02]	6.3270e-01[8.5e-02] [†]	4.9137e-01[1.1e-01] [†]	6.4794e-01[7.0e-02] [†]	8.4771e-01[6.9e-02]
1,2,3,4,7,8	9.0153e-01[4.8e-02]	9.2514e-01[3.9e-02]	7.3131e-01[7.6e-02] [†]	5.6765e-01[9.3e-02] [†]	6.9834e-01[6.7e-02] [†]	9.3598e-01[4.9e-02]
1,2,4,5,6,7	8.7708e-01[6.2e-02] [†]	9.1383e-01[7.7e-02]	6.6460e-01[5.3e-02] [†]	5.3615e-01[9.3e-02] [†]	7.3748e-01[8.8e-02] [†]	9.4422e-01[5.4e-02]
1,2,4,5,6,8	8.6552e-01[5.3e-02]	8.9480e-01[5.5e-02]	6.7689e-01[8.1e-02] [†]	5.5982e-01[9.2e-02] [†]	7.2025e-01[1.1e-01] [†]	8.8305e-01[6.5e-02]
1,2,4,5,7,8	8.5448e-01[5.5e-02] [†]	8.8479e-01[5.8e-02] [†]	6.6040e-01[1.2e-01] [†]	5.6747e-01[8.4e-02] [†]	6.7246e-01[8.4e-02] [†]	9.3267e-01[4.4e-02]
1,3,4,5,6,7	7.8138e-01[5.3e-02]	8.1493e-01[7.3e-02]	6.0530e-01[7.7e-02] [†]	4.8095e-01[1.1e-01] [†]	6.2493e-01[5.4e-02] [†]	8.1943e-01[8.4e-02]
1,3,4,5,6,8	9.0486e-01[4.4e-02] [†]	8.8170e-01[5.2e-02] [†]	7.4689e-01[7.6e-02] [†]	6.1929e-01[5.2e-02] [†]	7.0891e-01[6.6e-02] [†]	9.4882e-01[2.4e-02]
1,3,4,5,7,8	8.8090e-01[6.5e-02] [†]	9.1154e-01[3.3e-02] [†]	7.3695e-01[8.0e-02] [†]	5.4450e-01[9.6e-02] [†]	7.0130e-01[9.3e-02] [†]	9.6901e-01[5.0e-02]
1,2,3,4,5,6,7	8.6779e-01[4.2e-02] [†]	8.9230e-01[3.8e-02]	6.9488e-01[5.6e-02] [†]	4.5881e-01[1.1e-01] [†]	7.0285e-01[7.0e-02] [†]	9.1499e-01[3.6e-02]
1,2,3,4,5,6,8	9.0109e-01[4.9e-02]	8.9839e-01[4.1e-02]	6.2210e-01[9.0e-02] [†]	4.7676e-01[1.3e-01] [†]	6.4757e-01[9.0e-02] [†]	9.2356e-01[3.9e-02]
1,2,3,4,6,7,8	8.0928e-01[4.7e-02] [†]	8.2673e-01[7.2e-02]	5.7374e-01[6.3e-02] [†]	4.4372e-01[8.3e-02] [†]	5.9273e-01[1.0e-01] [†]	8.7739e-01[3.9e-02]
1,2,3,5,6,7,8	8.9533e-01[3.8e-02] [†]	9.0948e-01[3.7e-02] [†]	6.7884e-01[9.4e-02] [†]	5.6640e-01[8.1e-02] [†]	6.8025e-01[7.2e-02] [†]	9.5091e-01[3.9e-02]
1,2,4,5,6,7,8	9.4277e-01[2.8e-02]	9.5227e-01[4.7e-02]	6.8811e-01[6.6e-02] [†]	5.7719e-01[1.3e-01] [†]	7.0919e-01[1.2e-01] [†]	9.8345e-01[3.7e-02]
1,3,4,5,6,7,8	8.7669e-01[5.6e-02]	8.5951e-01[6.2e-02]	6.4670e-01[7.6e-02] [†]	5.3246e-01[7.4e-02] [†]	7.0362e-01[1.0e-01] [†]	8.7981e-01[6.6e-02]
2,3,4,5,6,7,8	8.6804e-01[5.5e-02] [†]	9.4451e-01[5.5e-02]	6.6284e-01[8.6e-02] [†]	5.4698e-01[9.7e-02] [†]	7.2543e-01[9.5e-02] [†]	9.3047e-01[3.4e-02]
1,2,3,4,5,6,7,8	8.1483e-01[7.5e-02] [†]	8.7137e-01[4.5e-02]	5.7484e-01[5.7e-02] [†]	4.0086e-01[8.5e-02] [†]	6.1434e-01[8.0e-02] [†]	9.0619e-01[4.6e-02]

“†” means that the result is significantly different from that of the PBEA at a 0.05 level by the Wilcoxon’s rank sum test.

Fig. 7. Normalized non-dominated solution set of (a)IBEA, (b)NSGA-II, (c)MOEA/D+WS, (d)MOEA/D+PBI, (e)MOEA/D+TCH, (f)PBEA on the MDLB-MR instance “1, 2, 3, 4, 5, 6, 7, 8”, as shown by parallel coordinates.

(1) Despite inevitable convergence rate reduction associated with the number of objectives increasing

(Giagkiozis & Fleming, 2015), Pareto-based EAs (e.g., NSGA-II) can outperform indicator-based (e.g., IBEA) or decomposition-based EAs (e.g., MOEA/D) on some many-objective problems with relatively low dimensions, such as 4-objective MDLB-MR introduced in this study.

- (2) Despite a number of advantages, e.g., providing higher selection pressure toward the Pareto front, compared with Pareto-based EAs, decomposition-based EAs encounter difficulties on real-world problems (e.g., MDLB-MR) in which the Pareto front geometry is often unknown and irregular. These difficulties are likely to arise from the fact that it is not straightforward to determine a good set of weights a priori for MDLB-MR (Wang et al., 2015). Additionally, TCH scalarizing function has better probability to find superior solutions compared to other scalarizing functions.
- (3) Developing hybrid EAs (e.g., PBEA) of synthesizing the Pareto and non-Pareto selection criteria tends to be promising. It is worth pointing out that the NPC population evolution of PBEA is implemented by using MOEA/D+TCH. The above observations thus further reveal the effectiveness of the exploration in producing competitive individuals in terms of the PC population evolution of PBEA.

6. Conclusion

In this paper, we studied the many-objective MDLB-MR problem, which has not been tackled previously in DLBP literature but is very close to the real remanufacturing situation. In the beginning, MDLB-MR is defined with its assumptions; and an illustrative example for MDLB-MR is represented. The optimal solutions based on TAOG and AND TPD are found by coding the proposed mathematical model with the LINGO 11.0 optimizer, respectively. The example shows that, in order to obtain potential benefits of using robot paralleling in MDL-MR, (i) TAOG is an appropriate task precedence relation representation scheme for MDLB-MR, and (ii) not only sequential tasks but also parallel tasks in TAOG can be considered. For this reason, a mathematical model based on TAOG and parallel disassembly is proposed to solve the problem optimally. Since MDLB-MR has been identified as NP-hard, PBEA is also developed to solve practical MDLB-MR problems, which have medium or large scales generally, in remanufacturing systems.

In order to investigate the behavior of different EAs on MDLB-MR, a total of 63 instances composed by 8 TAOG models of realistic EOL products from literature are generated as the benchmark. In the experimental studies, the performance of the PBEA based on TAOG is firstly investigated by comparing its results with those obtained by the PBEA based on AND TPD on 4 MDLB-MR instances. The analyses show that TAOG and parallel disassembly not only decrease the cycle time, but also dramatically reduce the energy consumption of MDL-MR. Moreover, we provided a comparative study of PBEA

and five representative EAs selected from the three classes of approaches of dealing with many-objective problems. The computational results are evaluated with HV metric, and comparison results demonstrate that PBEA outperform all the other EAs, especially the three versions of MOEA/D, in terms of convergence, extensivity and uniformity of Pareto-optimal solution set on most of the medium and large sized MDLB-MR instances. According the results, some insights with respect to the design of EAs for MDLB-MR are gained. In particular, our study reveals that a combination of non-Pareto based EAs with Pareto selection criterion seems to be promising on many-objective MDLB-MR. In addition, the effectiveness of key components in PBEA is also verified, including encoding/decoding scheme, solution initialization approach and problem-specific variation operators. It is worth noting that our main contribution in solution method is development of these components, and the main aim of this paper is to introduce MDLB-MR and parallel disassembly by using robot paralleling, to develop mathematical and metaheuristic approaches for MDLB-MR and to constitute a way for the future research.

One area for further investigation is to compare more representative EAs on some relatively high-dimensional many-objective MDLB-MR problems. In addition, since more and more uncertain issues emerged in both the disassembly tools and the EOL products, in the subsequent work, we will study how to model the stochastic/fuzzy/interval disassembly task operation times and develop new mathematical formulations and many-objective approaches to deal with uncertainty in MDLB-MR.

Acknowledgements

This work was supported by the National Natural Science Foundation Committee of China under Grant No.51475347, the Major Project of Technological Innovation Special Fund of Hubei Province of China under Grant No.2016AAA016, and the Engineering and Physical Sciences Research Council of U.K. under Grant EP/N018524/1.

References

- Agrawal, S., & Tiwari, M. K. (2008). A collaborative ant colony algorithm to stochastic mixed-model u-shaped disassembly line balancing and sequencing problem. *International Journal of Production Research*, 46, 1405–1429.
- Altekin, F. T., & Akkan, C. (2012). Task-failure-driven rebalancing of disassembly lines. *International Journal of Production Research*, 50, 4955–4976.
- Altekin, F. T., Kandiller, L., & Ozdemirel, N. E. (2008). Profit-oriented disassembly-line balancing. *International Journal of Production Research*, 46, 2675–2693.

- Aydemir-Karadag, A., & Turkbey, O. (2013). Multi-objective optimization of stochastic disassembly line balancing with station paralleling. *Computers and Industrial Engineering*, 65, 413–425.
- Bartholdi, J. J. (1993). **Balancing Two-sided Assembly Lines: A Case Study**. *International Journal of Production Research*, 32, 2447–2461.
- Battaïa, O., & Dolgui, A. (2013). A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics*, 142, 259–277.
- Bentaha, M. L., Battaïa, O., & Dolgui, A. (2015b). **An exact solution approach for disassembly line balancing problem under uncertainty of the task processing times**. *International Journal of Production Research*, 53, 1807–1818.
- Bentaha, M. L., Battaïa, O., Dolgui, A., & Hu, S. J. (2015c). **Second order conic approximation for disassembly line design with joint probabilistic constraints**. *European Journal of Operational Research*, 247, 957–967.
- Bentaha, M. L., Dolgui, A., & Battaïa, O. (2015a). **A bibliographic review of production line design and balancing under uncertainty**. *IFAC-PapersOnLine*, 48, 70–75.
- Bentaha, M. L., Dolgui, A., Battaïa, O., Riggs, R. J., & Hu, J. (2018). **Profit-oriented partial disassembly line design: dealing with hazardous parts and task processing times uncertainty**. *International Journal of Production Research*, (pp. 1–23).
- Boysen, N., Flidner, M., & Scholl, A. (2007). **A classification of assembly line balancing problems**. *European Journal of Operational Research*, 183, 674–693.
- Bullinger, H. J., & Ammer, E. D. (1984). **Computer-aided depicting of precedence diagrams A step towards efficient planning in assembly**. *Computers and Industrial Engineering*, 8, 165–169.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6, 182–197.
- Dimitriadis, S. G. (2006). **Assembly Line Balancing and Group Working: A Heuristic Procedure for Workers' Groups Operation on the Same Product and Workstation**. *Computers and Operations Research*, 33, 2757–2774.
- Ding, L. P., Feng, Y. X., Tan, J. R., & Gao, Y. C. (2010). **A new multi-objective ant colony algorithm for solving the disassembly line balancing problem**. *International Journal of Advanced Manufacturing Technology*, 48, 761–771.

- Fattahi, P., Roshani, A., & Roshani, A. (2011). A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem. *The International Journal of Advanced Manufacturing Technology*, 53, 363–378.
- 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, 1261–1278.
- Giagkiozis, I., & Fleming, P. J. (2015). Methods for multi-objective optimization: An analysis. *Information Sciences*, 293, 338–350.
- Güngör, A., & Gupta, S. (2001). A solution approach to the disassembly line balancing problem in the presence of task failures. *International Journal of Production Research*, 39, 1427–1467.
- Güngör, A., & Gupta, S. (2002). Disassembly line in product recovery. *International Journal of Production Research*, 40, 2569–2589.
- Hezer, S., & Kara, Y. (2015). A network-based shortest route model for parallel disassembly line balancing problem. *International Journal of Production Research*, 53, 1849–1865.
- Homem, L. S., de Mello, & Sanderson, A. C. (1990). AND/OR graph representation of assembly plans. *IEEE Transactions on Robotics and Automation*, 6, 188–199.
- Ilgın, M. A., Akcay, H., & Araz, C. (2017). Disassembly line balancing using linear physical programming. *International Journal of Production Research*, 55, 6108–6119.
- Jiang, H., Yi, J. J., Chen, S. L., & Zhu, X. M. (2016). A multi-objective algorithm for task scheduling and resource allocation in cloud-based disassembly. *Journal of Manufacturing Systems*, 41, 239–255.
- Kalayci, C. B., & Gupta, S. M. (2013). A particle swarm optimization algorithm with neighborhood-based mutation for sequence-dependent disassembly line balancing problem. *The International Journal of Advanced Manufacturing Technology*, 69, 197–209.
- Kalayci, C. B., & Gupta, S. M. (2014). A tabu search algorithm for balancing a sequence-dependent disassembly line. *Production Planning and Control*, 25, 149–160.
- Kalayci, C. B., Hancilar, A., Güngör, A., & Gupta, S. M. (2015). Multi-objective fuzzy disassembly line balancing using a hybrid discrete artificial bee colony algorithm. *Journal of Manufacturing Systems*, 37, 672–682.
- Kalayci, C. B., Polat, O., & Gupta, S. M. (2016). A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem. *Annals of Operations Research*, 242, 321–354.

- Koc, A., Sabuncuoglu, I., & Erel, E. (2009). Two exact formulations for disassembly line balancing problems with task precedence diagram construction using an and/or graph. *IIE Transactions*, 41, 866–881.
- Lambert, A. J. D. (1997). Optimal disassembly of complex products. *International Journal of Production Research*, 35, 2509–2524.
- Lambert, A. J. D. (1999). Linear programming in disassembly/clustering sequence generation. *Computers and Industrial Engineering*, 36, 723–738.
- Lambert, A. J. D. (2007). Optimizing disassembly processes subjected to sequence-dependent cost. *Computers and Operations Research*, 34, 536–551.
- Li, B. D., Li, J. L., Tang, K., & Yao, X. (2015). Many-objective evolutionary algorithms: a survey. *ACM Computing Surveys (CSUR)*, 48, 1–35.
- Li, H., & Zhang, Q. F. (2009). Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE Transactions on Evolutionary Computation*, 13, 284–302.
- Li, M. Q., Yang, S. X., & Liu, X. H. (2016a). Pareto or non-pareto: Bi-criterion evolution in multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 20, 645–665.
- Li, Z. X., Tang, Q. H., & Zhang, L. P. (2016b). Minimizing energy consumption and cycle time in two-sided robotic assembly line systems using restarted simulated annealing algorithm. *Journal of Cleaner Production*, 135, 508–522.
- Liu, J. Y., Zhou, Z. D., Pham, D. T., Xu, W. J., Yan, J. W., Liu, A. M., Q., J. C., & Liu, Q. (2018). An improved multi-objective discrete bees algorithm for robotic disassembly line balancing problem in remanufacturing. *The International Journal of Advanced Manufacturing Technology*, 97, 39373962.
- 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.
- McGovern, S. M., & Gupta, S. M. (2007). A balancing method and genetic algorithm for disassembly line balancing. *European Journal of Operational Research*, 179, 692–708.
- Mete, S., Çil, Z. A., Ağpak, K., Özceylan, E., & Dolgui, A. (2016). A solution approach based on beam search algorithm for disassembly line balancing problem. *Journal of Manufacturing Systems*, 41, 188–200.

- Mete, S., Çil, Z. A., Özceylan, E., Ağpak, K., & Battia, O. (2018). **An optimisation support for the design of hybrid production lines including assembly and disassembly tasks.** *International Journal of Production Research*, (pp. 1–15).
- Nilakantan, J. M., Huang, G. Q., & Ponnambalam, S. G. (2015). An investigation on minimizing cycle time and total energy consumption in robotic assembly line systems. *Journal of Cleaner Production*, *90*, 311–325.
- Nilakantan, J. M., Li, Z. X., Tang, Q. H., & Nielsen, P. (2017). Multi-objective co-operative co-evolutionary algorithm for minimizing carbon footprint and maximizing line efficiency in robotic assembly line systems. *Journal of Cleaner Production*, *156*, 124–136.
- Özceylan, E., Kalayci, C. B., Güngör, A., & Gupta, S. M. (2018). Disassembly line balancing problem: a review of the state of the art and future directions. *International Journal of Production Research*, (pp. 1–23).
- Özceylan, E., & Paksoy, T. (2013). **Reverse supply chain optimisation with disassembly line balancing.** *International Journal of Production Research*, *51*, 5985–6001.
- Özceylan, E., & Paksoy, T. (2014). **Interactive fuzzy programming approaches to the strategic and tactical planning of a closed-loop supply chain under uncertainty.** *International Journal of Production Research*, *52*, 2363–2387.
- Paksoy, T., Güngör, A., Özceylan, E., & Hancilar, A. (2013). Mixed model disassembly line balancing problem with fuzzy goals. *International Journal of Production Research*, *51*, 6082–6096.
- Pistolesi, F., Lazzerini, B., Mura, M. D., & Dini, G. (2018). **EMOGA: a Hybrid Genetic Algorithm with Extremal Optimization Core for Multiobjective Disassembly Line Balancing.** *IEEE Transactions on Industrial Informatics*, *14*, 1089–1098.
- Prakash, P., Ceglarek, D., & Tiwari, M. K. (2012). **Contraint-based simulated annealing (CBSA) approach to solve the disassembly scheduling problem.** *International Journal of Advanced Manufacturing Technology*, *60*, 1125–1137.
- Riggs, R. J., Battaia, O., & Hu, S. J. (2015). **Disassembly line balancing under high variety of end of life states using a joint precedence graph approach.** *Journal of Manufacturing Systems*, *37*, 638–648.
- Roshani, A., & Giglio, D. (2017). Simulated annealing algorithms for the multi-manned assembly line balancing problem: minimising cycle time. *International Journal of Production Research*, *55*, 2731–2751.

- Tang, Y., Zhou, M. C., Zussman, E., & Caudill, R. (2002). Disassembly modeling, planning and application. *Journal of Manufacturing Systems*, 21, 200–217.
- Vongbunyong, S., & Chen, W. H. (2015). *Disassembly Automation: Automated Systems with Cognitive Abilities*. Sustainable production, life cycle engineering and management. Berlin : Springer.
- Wang, R., Purshouse, R. C., & Fleming, P. J. (2015). Preference-inspired co-evolutionary algorithms using weight vectors. *European Journal of Operational Research*, 243, 423–441.
- Xiao, S., Wang, Y., Yu, H., & Nie, S. (2017). An entropy-based adaptive hybrid particle swarm optimization for disassembly line balancing problems. *Entropy*, 19, 596.
- Y., R., Yu, D., Zhang, C., Tian, G., Meng, L., & Zhou, X. (2017). An improved gravitational search algorithm for profit-oriented partial disassembly line balancing problem. *International Journal of Production Research*, 55, 7302–7316.
- Yuan, Y., & Xu, H. (2015). Multiobjective flexible job shop scheduling using memetic algorithms. *IEEE Transactions on Automation Science and Engineering*, 12, 336–353.
- Zhang, Z., Wang, K., Zhu, L., & Wang, Y. (2017). A Pareto improved artificial fish swarm algorithm for solving a multi-objective fuzzy disassembly line balancing problem. *Expert Systems with Applications*, 86, 165–176.
- Zheng, F., He, J., Chu, F., & Liu, M. (2018). A new distribution-free model for disassembly line balancing problem with stochastic task processing times. *International Journal of Production Research*, (pp. 1–13).
- Zhu, L., Zhang, Z., & Wang, Y. (2018). A Pareto firefly algorithm for multi-objective disassembly line balancing problems with hazard evaluation. *International Journal of Production Research*, (pp. 1–21).
- Zitzler, E., & Künzli, S. (2004). Indicator-based selection in multiobjective search. *PPSN 2004: Parallel Problem Solving from Nature - PPSN VIII*, (pp. 832–842).
- Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3, 257–271.