



# A Multi-objective Two-sided Disassembly Line Balancing Optimization Based on Artificial Bee Colony Algorithm: A Case Study of an Automotive Engine

Lei Zhang<sup>1</sup> · Yuanfeng Wu<sup>1,2</sup> · Xikun Zhao<sup>3</sup> · Shiwen Pan<sup>1,2</sup> · Ziqi Li<sup>1,2</sup> · Hong Bao<sup>4</sup> · Yongtin Tian<sup>1,2</sup>

Received: 19 November 2020 / Revised: 13 March 2021 / Accepted: 7 June 2021 / Published online: 29 October 2021  
© Korean Society for Precision Engineering 2021

## Abstract

Disassembly is an important part of green manufacturing and remanufacturing. The disassembly line is an optimum form for mass and automatic disassembly in the industry. To optimize the disassembly system and the use of resources, the disassembly line balancing problem (DLBP) has attracted much attention. Compared with the conventional one-sided straight disassembly line, the two-sided disassembly line can use both the left and right sides of a conveyor belt for disassembly operation, thereby improving production efficiency, especially for large-sized and complicated products. On the other hand, due to the constraints and precedence among parts, it is a challenge to plan the disassembly scheme for a two-sided disassembly line. In this paper, a model is established to solve a two-sided disassembly line balancing problem (TDLBP). First, a hybrid graph is utilized to express constraints and precedence relationships, and a novel encoding and decoding method is developed for the disassembly scheme planning of a two-sided line for handling the challenge caused by constraints and precedence among parts. Then, a multi-objective TDLBP optimization model is proposed including the number of mated-workstations, idle time, smoothness index, the auxiliary indicator, and a meta-heuristic based on an artificial bee colony (ABC) algorithm is designed to solve TDLBP. Finally, the proposed model and method are applied to an automotive engine case, and the results are compared with NSGA-II, hybrid artificial bee colony algorithm (HABC), and multi-objective flower pollination algorithm (MOFPA) to verify the practicality of the proposed model in solving the TDLBP.

**Keywords** Two-sided disassembly line balancing · Improved multi-objective artificial bee colony · Multi-objective optimization · Sustainable manufacturing

## 1 Introduction

The rapid development of science and technology has accelerated the speed of product updates but also has caused a large number of discarded products such as computers, cell

phones, LCDs, and vehicles. These end-of-life products contain a lot of resources such as metals, glass, and some toxic substances. It is essential to recycled these discarded products properly, not only to save resources but also to protect the environment. With the improvement of environmental protection awareness, there is an increasing interest in recycling, reuse, and remanufacturing technologies. And the recycling, reusing, remanufacturing of end-of-life products have attracted widespread concern [1]. Disassembly refers to a systematic process of extracting each part from a product for recycling and reuse and is a critical part of product recovery and sustainable manufacturing [2].

Disassembly can be performed in a disassembly cell, a single workstation, or a disassembly line. The disassembly line is a flow line designed to disassemble parts from a product automatically for subsequent recovery processes and remanufacturing, and the conventional disassembly line configuration is shown in Fig. 1a. It has many advantages

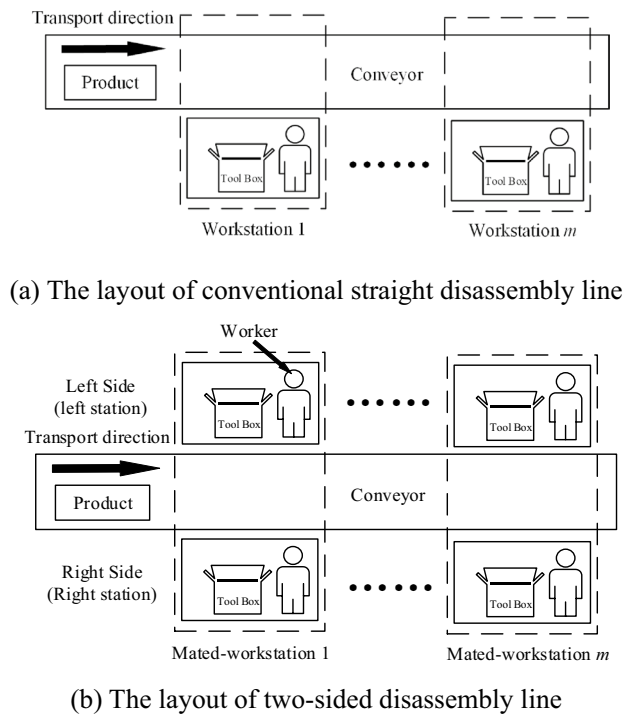
✉ Lei Zhang  
zhlei@hfut.edu.cn

<sup>1</sup> School of Mechanical Engineering, Hefei University of Technology, Hefei, China

<sup>2</sup> Key Laboratory of Green Design and Manufacturing of Mechanical Industry, Hefei University of Technology, Hefei, China

<sup>3</sup> State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing, China

<sup>4</sup> Industrial Training Center, Hefei University of Technology, Hefei, China



**Fig. 1** The layout of two-sided disassembly

compared with the first two disassembly modes. For example, it is the most suitable setting for disassembly of large products or small products in large quantities; the disassembly line offers the highest disassembly efficiency, and it is widely valued as the best way to automatically disassemble parts in remanufacturing especially in industrial mass production [3, 4].

To improve the disassembly system and optimize the use of resources, the researchers proposed the disassembly line balancing problem (DLBP), that is, to reasonably assign disassembly tasks to each workstation, evenly distribute tasks, and achieve a balancing state while satisfying the constraint and precedence relationship among parts [5]. Recently, DLBP has attracted attention in industry and academy.

DLBP is a complicated combinatorial optimization problem. Different types of disassembly lines are different in the disassembly processing method, and the mathematical models of evaluation indicators are established based on it [6]. In previous studies, such as disassembly efficiency, the number of workstations, cost, hazard degree, demand index are frequently selected indicators [7, 8]. To reduce construction costs and improve disassembly efficiency, the number of workstations and the idle time are the preferred optimization objectives. Considering the meaning of DLBP aforementioned, it is necessary to take the smoothness of the workload distribution of each workstation into account,

namely, the smoothness index [4]. In addition, it is vital to minimize the number of direction changes and tool changes (called auxiliary operation indicator) to simplify auxiliary operations. Therefore, the following four objectives are considered in this paper: (1) minimize the number of mated-workstations; (2) minimize the idle time; (3) minimize the smoothness index; (4) minimize the auxiliary indicator.

In order to address DLBP, many researchers developed some heuristic methods to get optimal or near-optimal results of DLBP. For instance, Ren et al. [9] used a 2-opt algorithm; Ilgin [10] constructed a DEMATEL-based heuristic. Some researchers have adopted mathematical programming methods in DLBP, such as a mixed-integer linear programming model [11–13]. DLBP is an NP-complete problem, so the amount of computation increases exponentially as the structure of a product becomes complex [14]. Although mathematical programming is suitable for solving simple examples of DLBP, due to its combinatorial nature, it will soon fail to solve large-sized products [15]. On the contrary, the meta-heuristic methods can get satisfactory results within a reasonable time. It has been widely used and has become the mainstream method in DLBP, such as particle swarm optimization (PSO) [16], ant colony optimization (ACO) [17, 18], genetic algorithm (GA) [13, 14, 19], gravitational search algorithm [20], discrete bee algorithm (DBA) [21], fruit-fly optimization algorithm (FOA) [22]. When dealing with various objectives, some researchers consider each objective lexicographically, i.e., optimize each objective one by one [14–17]. But this is essentially a single-objective perspective, which cannot fully consider each objective [23]. Besides, some studies integrated different objectives into one objective by weighting [13, 21, 24, 25]. Since the different objective units, it is difficult to remove the unit among different objective functions and set weight coefficients reasonably. Besides, scoring will bring subjectivity in the optimization process. Once the importance of each objective changes, the weights need to be changed and re-optimized. In addition, since the objectives may conflict with each other, it is hard for each objective to achieve the best value at the same time. On the contrary, the Pareto non-dominated relationship can be used to measure each objective comprehensively and is widely used in multi-objective optimization, and it is used in this paper.

All work mentioned above has made great achievements in the conventional DLBP which is mainly based on the straight disassembly line. The conventional straight disassembly line follows a simple linear process mode where workers can only disassemble one part at a time, and only one side is utilized, and the majority of current DLBP researches are based on it [7]. But this layout is less productive. There are different types of disassembly

lines. Except for the straight disassembly line, there are also line layouts such as two-sided disassembly lines. In a two-sided disassembly line, different disassembly tasks are performed on the same product in parallel at both sides of the line, and a pair of two facing workstations crossing the line is called a “mated-workstation” (see Fig. 1b). A two-sided production line has some advantages, such as higher disassembly efficiency, shorter disassembly line length, and lower construction and maintenance cost [26]. Similarly, the two-sided disassembly line needs to balance various resource utilization, and the DLBP also exists in the two-sided disassembly line, namely two-sided DLBP (TDLBP), regarded as an extension of conventional DLBP. However, due to the constraint relationship among parts, it is a bit difficult to coordinate asynchronous disassembly operations at both sides. This is the main challenge of two-sided disassembly scheme planning. Therefore, it is meaningful to research the TDLBP. In terms of optimization algorithm, Karaboga and Basturk [27] introduced an optimization algorithm based on bee colony behavior known as the artificial bee colony (ABC) algorithm. Due to the fewer control parameters, it is easier to implement and has an excellent optimization performance [28]. It has already been applied to conventional DLBP, and it stands out from other algorithms [15, 29]. In recent years, TDLBP has started to be studied. Wang et al. [30] first introduced the concept of TDLBP and used a multi-objective flower pollination algorithm for optimization. Recently, Kucukkoc [13] used an improved genetic algorithm for the optimization of TDLBP. To date, there are few studies on the TDLBP. This paper proposed a model including a novel encoding, decoding method, and an improved multi-objective artificial bee colony (IMOABC) algorithm to solve TDLBP.

Compared with the current DLBP researches, the novelty and main contributions of this paper can be summarized as follows.

1. This paper describes the product constraint and connect relationships based on the hybrid graph and defines the two matrices to digitize the constraint relationships for programming.
2. A two-sided line disassembly sequence model and a set of coding and decoding methods are proposed.
3. A multi-objective two-sided disassembly line model is established to optimize the number of mated-workstations, idle time, smoothness, and auxiliary processes, and the ABC algorithm is adapted and applied to the current model.

Compared with Kucukkoc’s work [13], we use a hybrid graph model instead of the AND/OR graph model. In the optimization process, we based on the Pareto non-dominated relationship instead of weighting, which can avoid the conflict of each objective. Once the demand changes, we can simply reselect a solution from the obtained Pareto set, whereas Kucukkoc’s method requires reconfiguring coefficients of the objective function and re-optimizing it.

The rest of the paper is as follows. In Sect. 2, a hybrid graph is used to present precedence and connection of product structure description, as well as a disassembly scheme planning method, including the disassembly sequence encoding and decoding method. The multi-objective optimization function and the IMOABC algorithm are represented in Sect. 3. The performance of the IMOABC and the practicality of the proposed model are verified in Sect. 4. Section 5 introduces conclusions and future works.

## 2 Precedence Model and Disassembly Scheme Generation

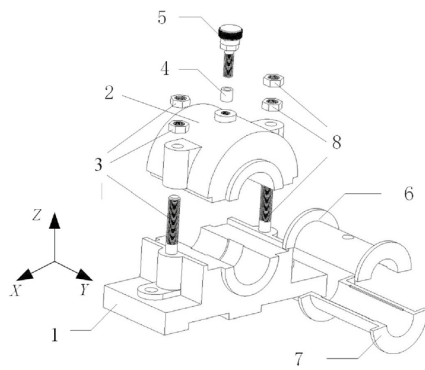
### 2.1 Constraints and Precedence Relationship Illustration

To illustrate the product structure and determine the order of disassembly tasks, some information models are used such as the directed graph, AND/OR graph, and Petri net [31, 32]. These graph models only represent the disassembly priority relationship among parts, but the actual connection relation that helps designers to understand complicated structures among parts cannot be illustrated [33]. The hybrid graph model can compensate for it and express direct and indirect precedence relation intuitively [24]. So, the hybrid model is proposed for product structure description. A bearing model and its hybrid graph are represented in Fig. 2.

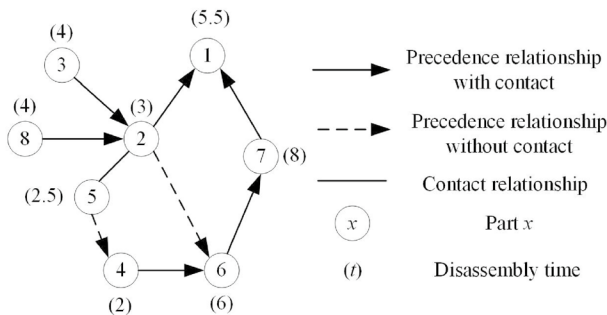
Then, the disassembly sequence can be generated by analyzing these information models, which is important for DLBP. To facilitate programming, a connection matrix  $M_c = [c_{ij}]_{n \times n}$  and a precedence matrix  $M_p = [p_{ij}]_{n \times n}$  are introduced to describe the connection and precedence relationship of part  $i$  and  $j$  of a product, and the meanings of  $c_{ij}$  and  $p_{ij}$  are shown in Eqs. (1) and (2).

$$c_{ij} = \begin{cases} 1 & \text{if part } i \text{ and } j \text{ connect each other} \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

$$p_{ij} = \begin{cases} 1 & \text{if part } i \text{ constrains part } j \\ 0 & \text{otherwise} \end{cases}. \quad (2)$$



(a) The exploded view of a bearing



(b) Hybrid graph

Fig. 2 The exploded view and hybrid graph of a certain sliding bearing

The  $M_c$  and  $M_p$  of the bearing in Fig. 2 are as follows:

$$M_c = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M_p = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Similarly, the  $M_c$  and  $M_p$  of any product can be obtained by this method of analysis, which possesses generality.

**Assumptions** The disassembly task assignment method and following multi-objective models are based on these assumptions for problem simplification [14, 34]: (a) The disassembly time of each part is known in advance; (b) Completely disassemble the product and disassemble a single product on the disassembly line; (c) The supply of end-of-life product is infinite; (d) Each task can only be assigned to one station; (e) The total disassembly time of tasks assigned to one side of mated-workstation shall not exceed cycle time  $CT$ ; (f) All the parts in a product are available and recyclable, without deletions, additions, modifications, physical or functional defects.

### 2.2 Disassembly Sequence Generation of Two-sided Disassembly Line

In order to disassemble a part (such as part  $g$ ) from a product, the following conditions (i.e. *constraint condition*) should be observed: part  $g$  is unrestricted by other parts and part  $g$  connects at most one part. As shown in Eq. (3), this constraint condition can be denoted by the elements of  $M_c$  and  $M_p$ :

$$\sum_{i=1}^n c_{i,g} \leq 1, \sum_{i=1}^n p_{i,g} = 0. \tag{3}$$

The connection and constraint relationship of a product changed once a part is disassembled. If the part  $g$  is disassembled, the  $g$ -th row of  $M_p$ , and the  $g$ -th row and  $g$ -th column elements of  $M_c$  must be cleared, i.e. *update*  $M_c$  and  $M_p$ . Iterate this procedure, a disassembly sequence can be obtained, and it can be used in conventional DLBP. Due to the constraints and precedence among parts and the asynchronous process nature of the two-sided disassembly, this conventional disassembly sequence cannot contain enough information for disassembly scheme planning. So, a modified disassembly sequence is proposed, which is expressed as  $X = [X^1, X^2]$ .

$X^1 = [p_1, \dots, p_n]$  is the same as the conventional disassembly sequence, where  $p_i$  ( $p_i \in P$ ,  $P$  is the parts set) corresponds to a specific part or task.  $X^2 = [s_1, \dots, s_n]$  is a sub-vector of  $X$  which indicates the disassembly side assignment result of each disassembly task. For example, if  $s_i = R$  (or  $L$ ), it means  $p_i$  will be disassembled at the right side (or left side) of the two-sided line. A feasible disassembly sequence generation method is described as follows:

Step 1: Construct the  $M_c$  and  $M_p$ , and two empty vectors  $X^1$  and  $X^2$ ;

Step 2: Based on Eq. (3), find all the parts that can be disassembled in this stage, and randomly select one;

Step 3: Insert the selected part into the corresponding empty position of  $X^1$ ;

Step 4: Update  $M_c$  and  $M_p$ ; If all the parts have been disassembled, go to Step 5, else go to Step 2;  
 Step 5: Fill  $X^2$  up with character ‘R’ or ‘L’ randomly;  
 Step 6: End.

### 2.3 Decoding of Two-sided Disassembly Scheme

The decoding refers to plan a disassembly scheme based on the type of the disassembly line and disassembly sequence, i.e. assigning each disassembly task to a certain workstation and determining its start and end time. It is different from the conventional one-sided disassembly line, because the disassembly scheme is an asynchronous process of a two-sided disassembly line, and the constraint relationship must be considered for operation coordination, which complicates the calculation of work time  $ST$ . In the job-shop scheduling problem, a job is assigned to a specific workstation depending on its start and end time [35]. This idea is applied to TDLBP for disassembly scheme planning in this paper, and three basic equations are proposed for decoding as shown in Eqs. (4) to (6).

$$t_{e,i} = t_{s,i} + t_{d,j}, j \in P, i = \begin{cases} R & \text{right side} \\ L & \text{left side} \end{cases}, \quad (4)$$

$$t_e = \max_{i=L,R} (t_{e,i}), t_e \leq CT, \quad (5)$$

$$ST_{ij} = \sum_{v=1}^c t_{d,v}^{i,j}, i = \begin{cases} R & \text{right side} \\ L & \text{left side} \end{cases}, \quad (6)$$

where,  $t_{s,i}$  and  $t_{e,i}$  respectively represent the start and end time of side  $i$ , and  $t_e$  is the end time judging whether to open a new workstation,  $CT$  is the cycle time (the maximum disassembly time).  $t_{d,j}$  is the disassembly time of part  $j$ . Equation (4) and (5) are used to determine whether the part can be disassembled in the current mated-workstation. Equation (6) is for computing the actual disassembly time of a mated-workstation, where  $c$  is the number of parts disassembled at side  $i$  of mated-workstation  $j$ ,  $t_{i,j}^{d,v}$  is the part  $v$ 's disassembly time assigned to side  $i$  of mated-workstation  $j$ .  $ST_{ij}$  is the sum disassembled time of side  $i$  of mated-workstation  $j$ . Note that  $i=R$  (right station) or  $L$  (left station). Then, the actual work time of a mated-workstation can be calculated, and the detailed decoding method is proposed in Fig. 3. Where  $P_m$  refers to the set of parts disassembled at mated-workstation  $m$ ;  $p_{m,L}$  and  $p_{m,R}$  refer to the last parts disassembled at the left and right side of mated-workstation, respectively;  $m$  is the number of mated-workstations;  $k$  is the iteration counter; the symbol “ $\rightarrow$ ” refers to the constraint,

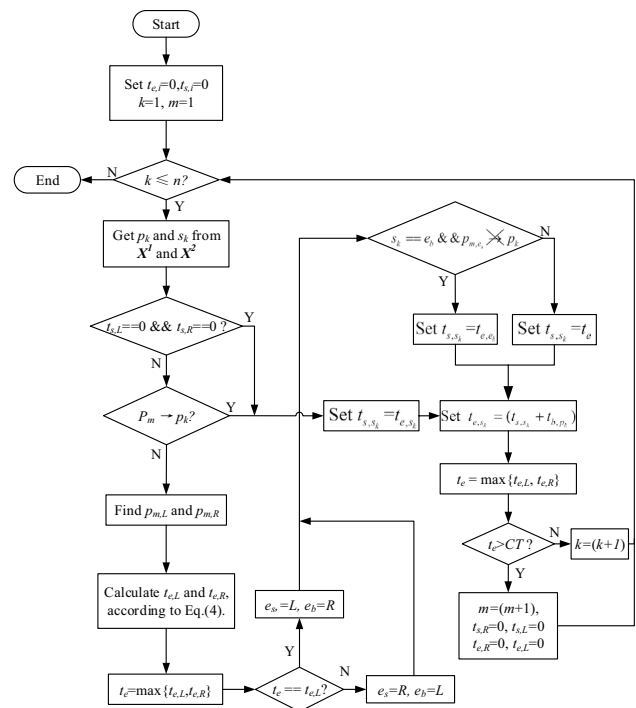
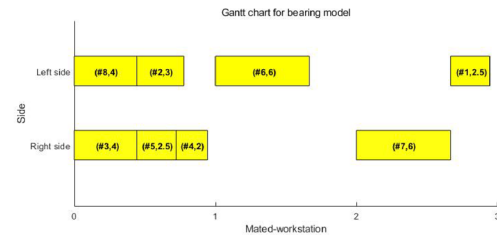
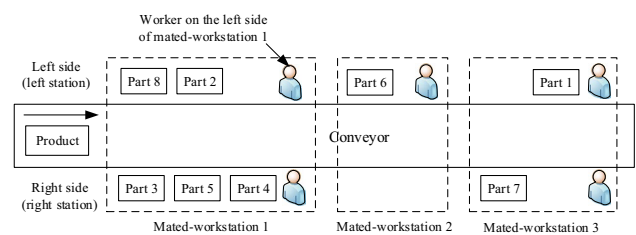


Fig. 3 The flow chart of the decoding process



(a). The Gantt chart of example sequence  $X$



(b). The disassembly scheme of example sequence  $X$

Fig. 4 The disassembly assignment scheme and the layout of the two-sided disassembly line

e.g. if  $p_1$  constrains  $p_2$ , it is denoted as “ $p_1 \rightarrow p_2$ ”. If  $p_1$  does not constrain  $p_2$ , it is expressed as “ $p_1 \not\rightarrow p_2$ ”.

Take a feasible solution  $X$  of Fig. 2s,  $X = [X^1 = [1-8], X^2 = [R,L,L,R,R,L,R,L]]$ , set cycle time  $CT=9$  s, and

disassembly time of each part is known (see Fig. 2b). Figure 4a is the Gantt chart to illustrate this scheme, and the two numbers in each block represent the part number and disassembly time. Figure 4b shows the task assignment and arrangement of this scheme example.

According to Eqs. (4) and (5) and the decoding method, the disassembly scheme can be planned. First, assign part 3 to the right side of mated-workstation 1. Since part 8 is not constrained by part 3, part 8 can be disassembled at the left side meanwhile. As part 3 and 8 both constrain part 2, part 2 can be dismantled after part 8 and 3 are removed (in the 4th second). The disassembly plans for part 5 and 4 are similar to the previous planning method. However, the mate-workstation 1 cannot have enough capacity for part 6's disassembly, so a new mated-workstation will open. As part 6 constrains part 7, part 7 will be disassembled after part 6's disassembly, i.e., in the 15th second. Since  $t_{s,2} + tR, 2d, 6 > 2 \times CT$ , part 7 must be disassembled at the right side of mated-workstation 3. No worker at this side because the right side of mated-workstation 2 is empty. According to the proposed decoding method, the tasks can be assigned to both sides considering constraints but the objective function is not determined, and the quality of the scheme cannot be measured.

### 3 Multi-objective Functions and Optimization Method

This section introduces a multi-objective model in Sect. 3.1, and Sect. 3.3 to 3.5 introduces an improved multi-objective artificial bee colony (IMOABC) algorithm for solving TDLBP. In the ABC algorithm, each food source is a feasible solution associated with an employed bee. Also, three kinds of bees are employed to search for food sources, namely, employed bee, onlooker bee, and scout bee. The employed bees search for new food sources (new solutions) and share information with onlooker bees. One onlooker bee evaluates the quality of food sources obtained by employed bees and select one for further exploration. If a food source is depleted, it will be abandoned and introduced into the scout bee, i.e., a new solution will be randomly generated.

#### 3.1 Multi-objective Functions

According to the available disassembly scheme, the balancing state of a disassembly line can be evaluated by the objective functions, including the number of mated-workstations ( $f_1$ ), idle time ( $f_2$ ), auxiliary operation indicator ( $f_3$ ), smoothness index ( $f_4$ ). Converted these objectives into minimization objectives for optimization, as shown in Eq. (7).

$$\min F = [f_1, f_2, f_3, f_4]. \quad (7)$$

(1) *Number of mated-workstations*. The number of workstations is an important indicator for evaluating the performance of the disassembly line. In a two-sided disassembly line, it is called "mated-workstation". Fewer mated-workstations can reduce construction costs, workers, equipment, and space occupation. This indicator is expressed in Eq. (8).

$$f_1 = m, \quad (8)$$

where  $m$  is the number of mated-workstation of a two-sided disassembly line, which is determined by the decoding process in Sect. 2.4.

(2) *Idle time*. In order to improve disassembly efficiency, the wasted time, i.e., idle time should be considered. The less the idle time, the higher the utilization rate of the disassembly system.

$$f_2 = \sqrt{\sum_{i=L}^{R'} \sum_{j=1}^m (CT - ST_{i,j})^2} \quad i = \begin{cases} L & \text{Left side} \\ R & \text{Right side} \end{cases}, \quad (9)$$

where  $ST_{i,j}$  is the disassembly time of mated-workstation  $j$  at side  $i$ ,  $m$  is the number of mated-workstations.

(3) *Auxiliary indicator*. Direction and tool change indicator is developed to measure the disassembly scheme's performance. Too many tool and direction changes will result in invalid operation and affect disassembly efficiency. The expression of the auxiliary indicator is shown in Eqs. (10) and (11).

$$f_3 = \sum_{i=1}^{r-1} (\alpha(d_i) + \beta(dt_i)) + \sum_{i=1}^{l-1} (\alpha(d_i) + \beta(dt_i)), \quad (10)$$

$$\alpha(d_i) = \begin{cases} 0 & \text{no direction change} \\ 1 & \text{if } 90^\circ \text{ change is required} \\ 2 & \text{if } 180^\circ \text{ change is required} \end{cases} \quad (11)$$

$$\beta(dt_i) = \begin{cases} 0 & \text{no tool change} \\ 1 & \text{if tool change is required} \end{cases}$$

where  $l$  is the number of tasks assigned to the left stations, and  $r$  is the number of tasks assigned to the right stations.  $d_i$  is the disassembly direction of part  $i$ ,  $dt_i$  is the disassembly tool of part  $i$ .  $\alpha$  and  $\beta$  are direction change operator and tool change operator respectively, and their meanings are shown in Eq. (11).

(4) *Smoothness index*. The smoothness index is used to measure the balance of workload distribution, which can improve disassembly efficiency and reduce the difference in workload assignment. Its prototype is proposed in [36] and modified to a two-sided line version based on the model proposed in this paper, as shown in Eq. (12).

$$f_4 = -\ln(2 \cdot m) / \left( \sum_{j=1}^m \sum_{i=R'}^{L'} \left( \frac{(CT - ST_{ij})^2}{\sum_{i=1}^m \sum_{j=R'}^{L'} (CT - ST_{ij})^2} \cdot \ln \left( \frac{(CT - ST_{ij})^2}{\sum_{i=1}^m \sum_{j=R'}^{L'} (CT - ST_{ij})^2} \right) \right) \right), i = \begin{cases} L \text{ Left side} \\ R \text{ Right side} \end{cases} \tag{12}$$

(5) Constraints.

$$m_{\min} = \left\lfloor 0.5 \times \left\lceil \frac{\sum_{j=1}^n t_{d,j}}{CT} \right\rceil \right\rfloor \leq m \leq n, \tag{13}$$

$$ST_{ij} \leq CT, i \in \{1, 2\}, j \in \{1, \dots, m\}, \tag{14}$$

$$\sqrt{\sum_{i=R'}^{L'} \sum_{j=1}^{m_{\min}} (CT - ST_{ij})^2} \leq f_2 \leq \sqrt{\sum_{i=R'}^{L'} \sum_{j=1}^{m_{\max}} (CT - ST_{ij})^2}, \tag{15}$$

$$j \in \{1, \dots, m\}, i = \begin{cases} L \text{ Left side} \\ R \text{ Right side} \end{cases}$$

Equation (13) is a constraint implying the maximum and the minimum number of mated-workstations. According to the assumptions in Sect. 2.2, Eq. (14) ensures the maximum workload of each workstation. Equation (15) describes the minimum and maximum idle time of a mated-workstation.

### 3.2 Multi-objective Optimizer and Hypervolume Indicator

The multi-objective optimization refers to finding a solution vector  $x$  in decision space  $S$  without violating predetermined constraints so that the objective function vector  $F(\mathbf{Z}) = [f_1(\mathbf{Z}), f_2(\mathbf{Z}), \dots, f_u(\mathbf{Z})]$  can reach the optimal solution. The objective vector and constraints are shown in Eq. (16).

$$\begin{aligned} \min F(\mathbf{Z}) &= [f_1(\mathbf{Z}), f_2(\mathbf{Z}), \dots, f_u(\mathbf{Z})] \\ \text{s.t.} \begin{cases} g_i(\mathbf{Z}) \leq 0, i = 1, 2, \dots, q \\ h_j(\mathbf{Z}) = 0, j = 1, 2, \dots, r, \\ \mathbf{Z} \in S \end{cases} \end{aligned} \tag{16}$$

where,  $\mathbf{Z} = [z_1, \dots, z_n]$  is a solution vector,  $S$  is the feasible space of  $\mathbf{Z}$ ,  $F$  is the multi-objective vector;  $u$  is the number of objective functions,  $q$  is the number of inequality constraints, and  $r$  is the number of equality constraints.

In multi-objective optimization, the Pareto dominance relationship is often used for solution comparison. For example,  $\mathbf{Z}_1$  and  $\mathbf{Z}_2$  are two solution vectors. If they satisfy Eqs. (17) and (18),  $\mathbf{Z}_1$  dominates  $\mathbf{Z}_2$ , denoted as  $\mathbf{Z}_1 < \mathbf{Z}_2$ . By Pareto dominance relationship, all objectives can be considered, and each goal can be balanced in consideration of complicated conflicts.

$$f_i(\mathbf{Z}_1) \leq f_i(\mathbf{Z}_2), \forall i \in \{1, \dots, u\}, \mathbf{Z}_1 \in S, \mathbf{Z}_2 \in S, \tag{17}$$

$$f_i(\mathbf{Z}_1) < f_i(\mathbf{Z}_2), \exists i \in \{1, \dots, u\}, \mathbf{Z}_1 \in S, \mathbf{Z}_2 \in S. \tag{18}$$

Deb et al. [37] combined GA with Pareto grade (PG) sorting and crowding distance (CD) in selection. A set of solutions can be sorted into different PGs by Eqs. (19) and (20). The CD of  $\mathbf{Z}_p$  can be calculated by Eq. (19), where  $k$  is the number of non-dominated solutions. The CD values of the non-dominated set boundary are denoted as  $CD(\mathbf{Z}_1) = CD(\mathbf{Z}_k) = \infty$ . If  $\mathbf{Z}_1$  and  $\mathbf{Z}_2$  satisfy Eq. (20) or (21),  $\mathbf{Z}_1$  dominates  $\mathbf{Z}_2$ . This mechanism is the core of the tournament selection based on the Pareto relationship.

$$CD(\mathbf{Z}_p) = \sum_{i=1}^u \frac{f_i^{p+1} - f_i^{p-1}}{f_i^{\max} - f_i^{\min}}, p \in \{2, \dots, k - 1\}, \tag{19}$$

$$PG(\mathbf{Z}_1) < PG(\mathbf{Z}_2), \tag{20}$$

$$PG(\mathbf{Z}_1) = PG(\mathbf{Z}_2), CD(\mathbf{Z}_1) > CD(\mathbf{Z}_2). \tag{21}$$

But this mechanism cannot compare or evaluate the quality of Pareto non-dominated front. The hypervolume indicator is used to measure the quality of a non-dominated set, which measures how much of the objective space is dominated by an approximation set [38]. This indicator does not require information about the real Pareto-front, and it can be denoted as [39]:

$$HV(NDS, R_0) = Leb \left( \bigcup_{\mathbf{Z}_i \in NDS} [f_1(\mathbf{Z}_i), f_1(R_0)] \times \dots \times [f_u(\mathbf{Z}_i), f_u(R_0)] \right), R_0 \in S, \mathbf{Z}_i < R_0, \tag{22}$$

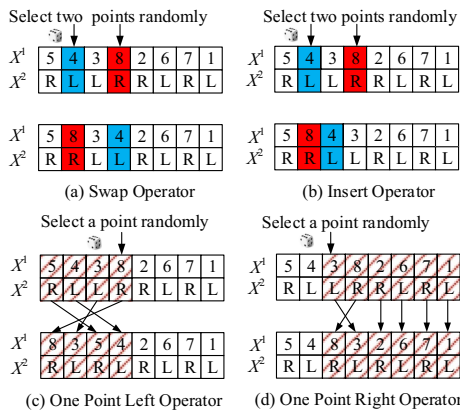


Fig. 5 The mechanism of these neighbor search operators

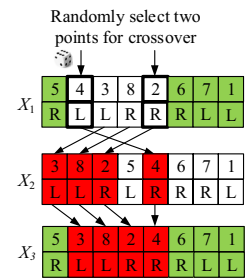
$$f_j(R_0) > f_j(Z_i), \forall j \in \{1, \dots, u\}, Z_i \in NDS, \quad (23)$$

where  $HV(NDS, R_0)$  is the size of the space covered by non-dominated set  $NDS$ ,  $R_0$  is a reference point,  $Leb$  is Lebesgue measure, and Eq. (23) is the constraint of Eq. (22). Note that  $R_0$  must be the same when comparing the optimization performance of different algorithms. Otherwise, the  $HV$  is not comparable.

Table 1 The frame and procedure of IMOABC

Input:	Iteration limit $IteLim$ , the population size $PopSize$ , crossover possibility $p_c$ , iteration counter $i = 1$
Output:	Employed bee set $\{X_j\}$
Start:	
(1)	Generated feasible solution set $\{X_j\}$ ( $j = 1, \dots, PopSize$ ) as <i>employed bee set</i> ;
(2)	$NDS(\{X_j\}) = non\_dominated\_sort(\{X_j\})$ , $Q = NDS(\{X_j\}) \cup Q$ ; // $NDS(\{X_j\})$ : non-dominated set of $\{X_j\}$ , and $NDS(\{X_j\})$ into $Q$
(3)	<b>While</b> $i \leq IteLim$ , <b>do</b>
(4)	$\{X_j'\} = local\_search\_loop(\{X_j\})$ ;
(5)	<b>If</b> $X_j < X_j'$ , <b>do</b> // $X_j'$ is the solution obtained from $X_j$ by neighborhood search
(6)	$Q = \{X_j'\} \cup Q$ ; // Append $\{X_j'\}$ into $Q$
(7)	<b>End if</b>
(8)	$\{Y_j\} = tournament\_selection(\{X_j'\})$ ; // get <i>onlooker bee set</i> $\{Y_j\}$ ;
(9)	Repeat row (4)-(8) on $\{Y_j\}$ , and get $\{Y_j'\}$ ;
(10)	$NDS(\{Y_j'\}) = non\_dominated\_sort(\{Y_j'\})$ , $Q = NDS\{Y_j'\} \cup Q$ ;
(11)	$Q' = two\_point\_crossover(Q, p_c\%)$ ;
(12)	$U = \{X_j'\} \cup \{X_j''\} \cup \{Y_j\} \cup \{Y_j'\} \cup Q$ , $NDS(U) = non\_dominated\_sort(U)$ ;
(13)	<b>If</b> $size(NDS(U) \cup Q') > PopSize$ , <b>do</b>
(14)	$\{X_j\} = non\_dominated\_sort(NDS(U) \cup Q')$ , ( $j = 1, \dots, PopSize$ );
(15)	<b>Else</b>
(16)	Randomly generate feasible solution set $\{S_j\}$ ; // $size(\{S_j\}) = PopSize - size(NDS(U) \cup Q')$
(17)	$\{X_j\} = \{S_j\} \cup NDS(U) \cup Q' \cup \{X_j\}$ , ( $j = 1, \dots, PopSize$ );
(18)	<b>End if</b>
(19)	$i + 1 \rightarrow i$ ;
(20)	<b>End while</b>

Fig. 6 Two-point crossover operator



### 3.3 Employed Bee Operator

In ABC, each employed bee (a feasible solution) shares information, and Kalayci and Gupta [40] applied two neighborhood search operators to DLBP, i.e., swap and insert operator. Similarly, in their subsequent studies, they proposed some operators for neighborhood search, such as swap operator, insert operator, one point left operator, one point right operator [40]. We adopted it and applied it to the modified disassembly sequence. As shown in Fig. 5, a feasible solution of Fig. 2 is taken as an example to introduce the mechanism of these operators. To explore solution space, a *local search loop* mechanism combining these four operators is proposed as follows for information sharing, where the determination of dominated relationship is based on Eqs. (17) to (21).



**Table 2** The name of automobile engine parts and properties

Number	Name	Quantity	Tool	Time/s	Direction
1	Camshaft cover bolt	12	T01	5 s	+z
2	Camshaft cover	1	–	3 s	+z
3	Camshaft cover sheet bolt	20	T02	4 s	+z
4	Camshaft cover sheet	8	–	2 s	+z
5	Cranking claw	1	T03	6.5 s	+x
6	Pulley	1	–	20 s	+x
7	Side cylinder head bolt	1	T04	6.5 s	+x
8	Side cylinder head	1	–	5.5 s	+x
9	Chain Constraints mechanism bolt	4	T04	4.5 s	+x
10	Chain Constraints mechanism	2	–	3 s	+x
11	Timing Chain	1	–	10 s	+z
12	Camshaft	1	–	3.2 s	+z
13	Upper cylinder head bolt	8	T05	4 s	+z
14	Upper cylinder head	1	T06	21 s	+z
15	Rocker arm component screw	16	T07	3 s	+z
16	Rocker arm	16	–	3 s	+z
17	Valve component	16	T08	4 s	+z
18	Spark plug	4	T09	22.5 s	+z
19	Cylinder bolt	10	T04	5.5 s	+z
20	Cylinder	1	T06	20 s	+z
21	Connecting rod cap	4	–	7 s	+z
22	Connecting rod component	4	–	10 s	+z
23	Connecting rod bolt	4	T05	7 s	+z
24	Crankshaft bearing cap bolt	10	T10	5 s	+z
25	Crankshaft bearing cap	5	–	3 s	+z
26	Crankshaft	1	–	20 s	+z
27	Upper cylinder	1	T06	23 s	+z
28	Flywheel nut	1	T11	15 s	-x
29	Flywheel bolt	6	T01	5.5 s	-x
30	Flywheel	1	–	30 s	-x
31	Oil cylinder foundation bolt	12	T01	5 s	+z
32	Oil cylinder foundation	1	–	6 s	+z

Step 1. Set the counter variable  $i=0$  and initialize local iteration limit  $t_{lim}$ ;

Step 2. Randomly select one of these four search operators above;

Step 3. Perform the selected operator on solution  $X$  and get new solution  $X'$ ;

Step 4. If  $i \leq t_{lim}$  and  $X$  dominates  $X'$ , set  $i=(i+1)$  and go to Step 3; else go to Step 4;

Step 5. Set  $X=X'$ ;

Step 6. Place  $X$  and  $X'$  in *employed bee set* and *new employed bee set* respectively. If  $X < X'$ , place  $X'$  in inferior set  $Q$ .

### 3.4 Onlooker Bee Operator

In traditional ABC, an onlooker bee selects a food source depending on the percent of nectar amount of each food

source among the total nectar amount. Here, the tournament selection method is adopted, i.e., choose two solutions and compare their PG and CD values, and select a better one according to the method introduced in Sect. 3.2. Then, use the *local search loop* mechanism to diversify the population. The onlooker bee colonies before and after neighbor search are noted as *onlook bees set* and *new onlooker bees set* respectively. If a solution is dominated by the solution before its neighbor search, put this solution into inferior set  $Q$ .

### 3.5 Scout Bee Operator

In the neighborhood search, the new solution  $X'$  may be dominated by the original solution  $X$ , and collect all the  $X'$ , namely inferior set  $Q$ . In this phrase, the two-point crossover of GA is adopted to explore solution space and perform a



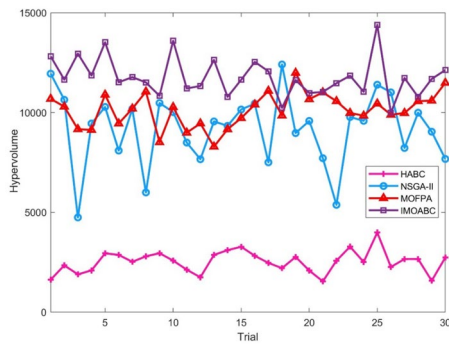


Fig. 11 The hypervolume values of listed methods for 30 trials

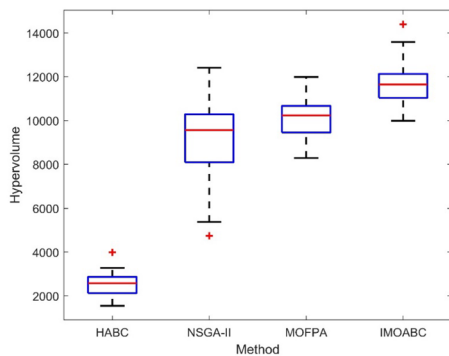


Fig. 12 Box plot of hypervolume value of four methods

obtained via this process. Of course, repeat this procedure until finding a feasible solution.

Finally, integrate employed bee colony, new employed bee colony, onlooker bee colony, new onlooker bee colony, and inferior set  $Q$  together, and rank them with non-dominated sorting. According to the ranking result, select a designated number of bees as the initial group for the next iteration until reach the iteration limit. Table 1 shows the pseudo-code of an improved multi-objective artificial bee colony (IMOABC) algorithm to solve TDLBP.

## 4 Case Study

In this section, in order to verify the practicality of the proposed two-sided model, including sequence coding, decoding method, and an optimization method, an automotive engine composed of 32 parts is taken as a case.

Some parts are simplified for simplification, e.g., belts, bearings, and pins are removed, and a group of fasteners is treated as one part. Some information on the engine parts

and their attributes, including name, quantity, disassembly tool, estimated disassembly time, and direction are shown in Table 2. The hybrid graph is shown in Figs. 7, and 8 is the explosive view of an engine. The connection matrix  $M_c$  and precedence matrix  $M_p$  are shown in Figs. 9 and 10. Also, Table 3 listed the essential disassembly tools. The two-sided disassembly schemes can be obtained and each objective is evaluated by the method and model introduced in Sects. 2 and 3.

### 4.1 Numerical Result and Algorithm Comparison

To test the optimization performance of IMOABC, three multi-objective algorithms are introduced for comparison, i.e., the famous NSGA-II [37], HABC [15] (an ABC recently applied to DLBP), and the multi-objective flower pollination algorithm (MOFPA) [30] (a recent TDLBP optimization method).

The parameters of IMOABC are set as follows: the population size  $PopSize = 50$ , iteration limit  $IteLim = 100$ , crossover possibility  $p_c = 0.8$ , local iteration limit  $t_{lim} = 5$ . The  $PopSize$  of NSGA-II, HABC, and MOFPA is the same as IMOABC. The population size of IMOABC, NSGA-II, and MOFPA is 50. According to the method introduced in Sect. 2, generate the disassembly sequence and two-sided disassembly scheme, and compute the value of each objective function in Sect. 3.1. In order to evaluate the solution quality obtained by each algorithm, the hypervolume indicator is used, and reference point  $R_0$  is set as (10,500,50,2).

Each algorithm runs 30 times, and Fig. 11 shows the hypervolume values of each run. It can be observed that the maximum hypervolume values of these four methods are 3991.7, 12,408.6, 11,988.7, 14,396.9, respectively, and IMOABC produces the best hypervolume result. To compare the stability of each algorithm, the hypervolume value box plot of 30 results for four algorithms is shown in Fig. 12. After removing the abnormal solution denoted in Fig. 12, the maximum hypervolume values of HABC, NSGA-II, MOFPA, IMOABC are 3279.8, 12,408.6, 11,988.7, 13,594.6, respectively. The best-known hypervolume result is obtained by IMOABC. From a comparison of Figs. 11 and 12, IMOABC can not only generate better solutions but also has better stability. Among these four algorithms, its overall performance is the best.

Table 4 and Fig. 13 show the Pareto non-dominated frontiers of the best hypervolume in each method. But due to the random assignment of ‘L’ and ‘R’ in  $X_2$ , it contains many different schemes with the same multi-objective value. The third column of Table 4 is the number of solutions with the same multi-objective value, and it is marked in Fig. 13.

**Table 4** The Pareto frontier of the four algorithms

Method	Solution	Same	$f_1$	$f_2$	$f_3$	$f_4$
HABC	1	1	6	178.835	36	1.509
	2	1	7	238.179	37	1.338
	3	1	7	236.489	40	1.302
	4	1	7	249.234	36	1.386
	5	1	8	279.548	34	1.212
	6	1	8	279.516	33	1.212
	7	1	8	278.895	37	1.218
	8	1	8	280.207	37	1.211
	9	1	8	291.061	32	1.269
	10	1	9	331.017	41	1.172
	11	1	9	321.377	37	1.175
	12	1	9	326.511	31	1.179
	13	1	10	356.319	33	1.119
NSGA-II	1	6	6	145.353	28	1.097
	2	14	7	200.909	29	1.079
	3	30	7	202.060	28	1.087
MOFPA	1	13	6	144.874	37	1.094
	2	29	6	156.055	27	1.213
	3	7	7	209.933	30	1.139
	4	1	8	261.712	30	1.112
IMOABC	1	14	6	145.291	27	1.100
	2	7	6	146.525	26	1.118
	3	29	6	147.938	24	1.139

Using the hypervolume indicator and box plot to evaluate the optimization results, whether in terms of the stability of the solution set or the quality of the solution set, IMOABC is the best.

In addition, it is necessary to analyze each objective function optimization result of the optimal Pareto front. As shown in Table 4 and Fig. 13, HABC, NSGA-II, MOFPA, and IMOABC can reach the best-known value of 6 on the objective *number of mated-workstations*, and also the objective *idle time* of four algorithms is 178.835, 145.353, 144.874 and 145.291. In terms of mated-workstation, compared with MOFPA and IMOABC, all methods can get the 6 mated-workstations, while HABC and NSGA-II have fewer solutions of six mated-workstations. The fewer mated-workstations and lower idle time mean less construction cost and higher work efficiency, and IMOABC performs well in these two objectives optimization.

Table 5 shows the comparison results of the best Pareto frontiers of these four methods in Fig. 13, it can be found that the number of mated-workstations of IMOABC is 6. But some schemes generated by other methods require 7, 8, or more mated-workstations. In terms of average idle time, compared with IMOABC, the disassembly schemes obtained by HABC, NSGA-II, and MOPFA increased by 90.85%, 31.54%, 10.75% respectively. It means that the optimization results of the IMOABC's Pareto solution on the idle time objective is significantly better than the other three methods, which improves work efficiency significantly. Moreover, a similar situation also occurs in the auxiliary indicator, which simplifies disassembly operation. However, its average smooth index is inferior to other methods except for HABC. Nevertheless, it is almost impossible to get the best solution for all the objectives, because the improvement of one objective may worsen

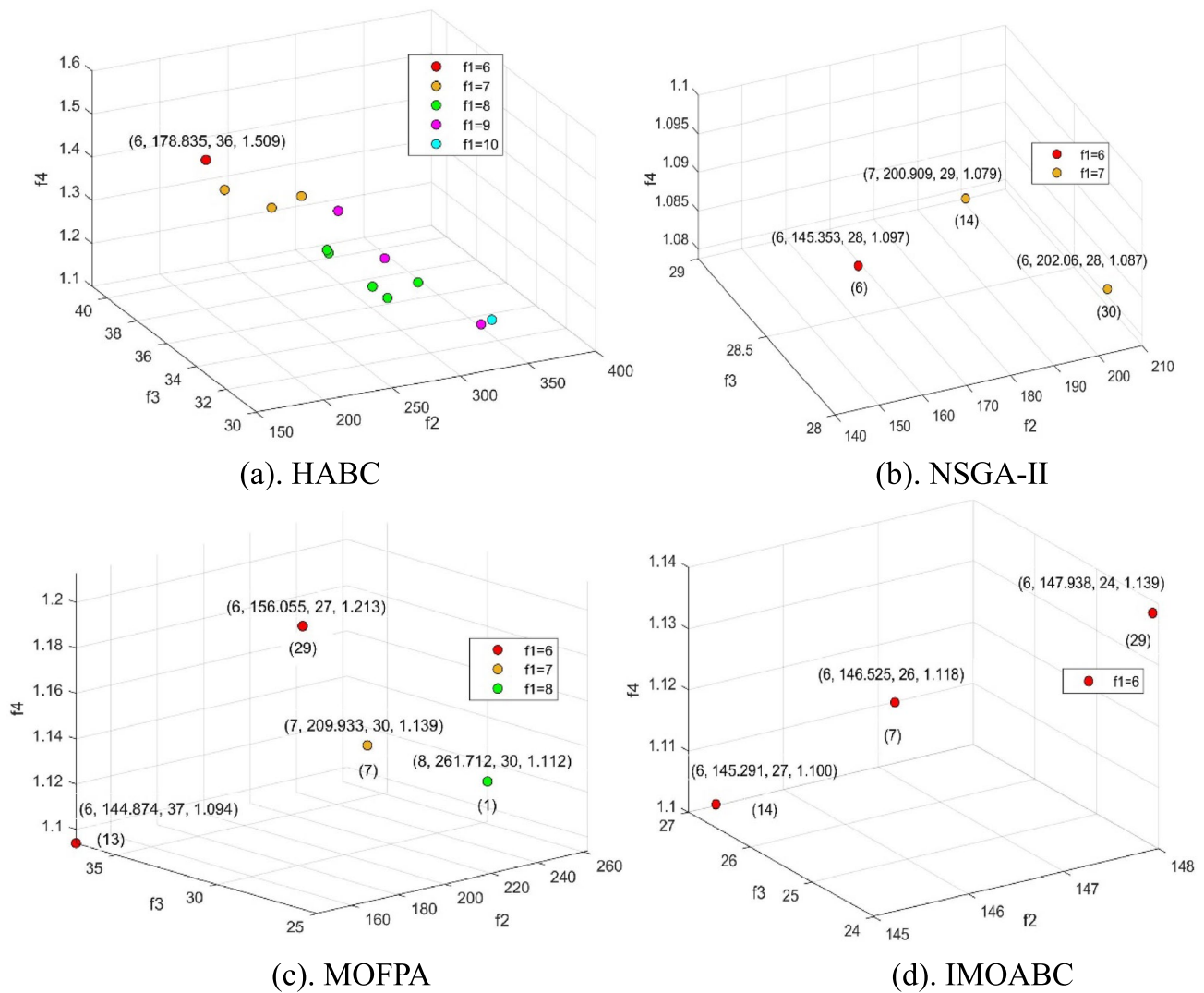


Fig. 13 The Pareto front of four methods

other objectives, and it is necessary to make a compromise on each objective.

Anyway, whether it is the quality of the Pareto solution set measured by the hypervolume indicator or the overall optimization effect of each objective, IMOABC performs better than the other three algorithms. In this case, the IMOABC has a better performance on two-sided disassembly line scheme optimization compared with these three methods.

### 4.2 Analysis of Two-sided Disassembly Schemes Selection

Take one disassembly sequence from each of the three Pareto solutions obtained by IMOABC, as shown in Table 6. After decoding, the Gantt charts of these three schemes are shown in Fig. 14 to illustrate the disassembly scheme of the two-sided disassembly line in detail. The parts disassembled from “+z”, “-x” and “+x” directions are represented

**Table 5** Comparison of the result of Fig. 13

Objective	Value	HABC	NSGA-II	MOFPA	IMOABC
$f_1$	6	1	6	49	50
	7	3	14	0	0
	8	5	30	1	0
	9	1	0	0	0
	10	1	0	0	0
$f_2$	max	356.319	202.0602	261.712	147.938
	avg	280.553	193.368	162.804	146.998
	min	178.835	145.3527	144.874	145.291
$f_3$	max	40	29	37	27
	avg	35.692	28.341	30.08	25.12
	min	31	28	27	24
$f_4$	max	1.509	1.097	1.213	1.139
	avg	1.254	1.086	1.170	1.125
	min	1.119	1.079	1.094	1.100
Number of non-dominated solutions		13	50	50	50

**Table 6** Three two-sided disassembly sequence

No	Part	Two-sided disassembly sequence
1	$X_1$	[5,29,6,28,30,1,2,3,18,15,7,8,16,9,10,11,4,12,13,14,19,17,20,24,21,25,23,26,22,27,31,32]
	$X_2$	[L,R,L,L,L,R,L,L,R,R,R,L,L,R,R,L,R,L,R,L,R,L,L,R,L,R,R,L]
2	$X_1$	[29,1,28,30,2,5,18,3,15,6,7,8,9,16,10,4,11,12,13,14,19,17,20,24,21,23,25,26,22,27,31,32]
	$X_2$	[R,R,L,L,L,R,L,R,L,R,L,R,L,R,R,L,L,R,L,R,L,R,L,L,R,R,L]
3	$X_1$	[29,28,1,30,2,18,3,5,15,6,7,8,16,9,10,11,4,13,12,14,17,19,20,21,24,23,25,26,27,22,31,32]
	$X_2$	[R,L,R,L,L,R,L,L,R,L,R,L,R,L,R,R,L,L,R,L,L,L,R,R,L,L,R,L,R,L]

by yellow, green and cyan color blocks on the Gantt chart respectively, and the block length represents the specific disassembly time of the parts.

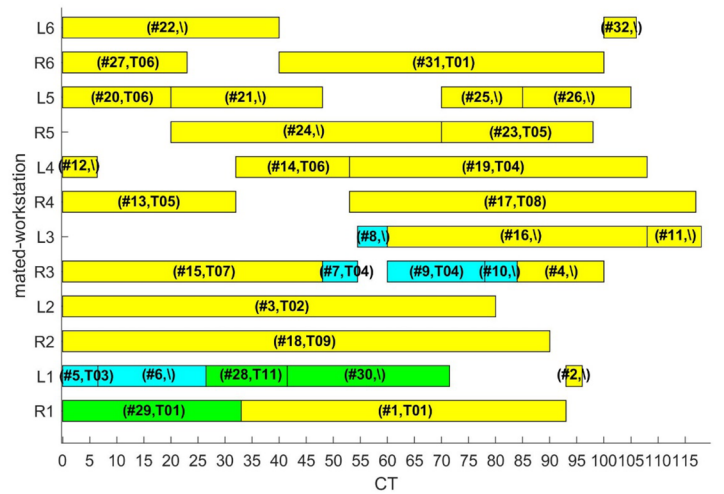
Note that the part number and tool number are marked on each block. It can be observed that the parts with the same direction and tool are concentrated, which reduces the auxiliary operations significantly.

**Table 7** The scale of scoring and description

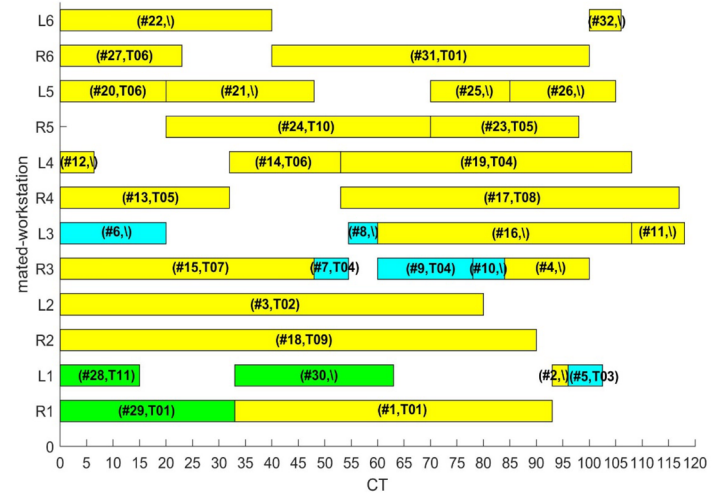
Value	Definition
1	Equal importance
3	$i$ is slightly important than $j$
5	$i$ is important than $j$
7	$i$ is stronger important than $j$
9	$i$ is absolute important than $j$
2,4,6,8	The median of two adjacent importance values
Reciprocal	The comparing result of $j$ and $i$ is the reciprocal of comparing $i$ and $j$

In the Pareto front, each solution does not dominate each other. The Pareto-based optimization process doesn't need to consider which objective is the most important in optimization, but only the dominant relationship between the objective vectors. The process is not mixed with any subjective factors, and the constraints and conflicts between objectives and be avoided. After obtaining the Pareto set, the importance of each objective depends on the subjective demand of decision-makers, and the solution that best meets the requirements can be selected from it. Analytical hierarchical process (AHP) is a critical multicriteria decision-making method that balances the interactions among decision criteria and widely used in decision making, and it is chosen for scheme selection [41]. The main steps of AHP: (1) remove the dimension of each objective; (2) compare the importance of each objective and scoring to get a judgment matrix; (3) transform multi-objectives into a single objective [42]. Equation (24) denotes the dimensionless equation, where  $\max f_i$  and  $\min f_i$  are the maximum and minimum values of  $f_i$  components

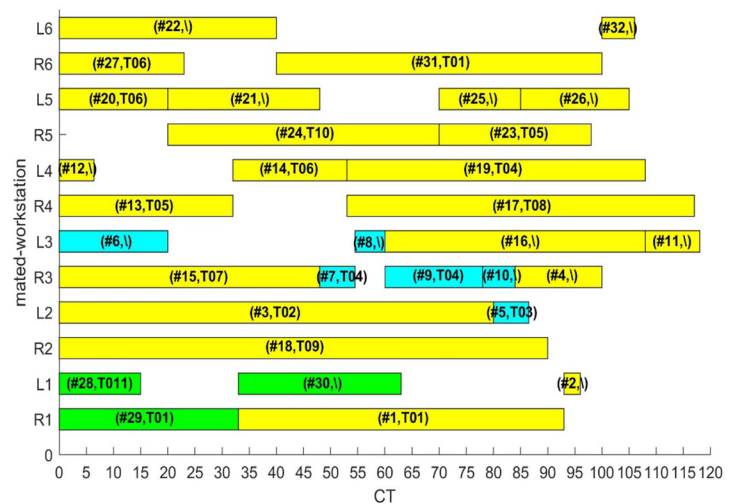
**Fig. 14** Three disassembly schemes of engine two-sided disassembly line



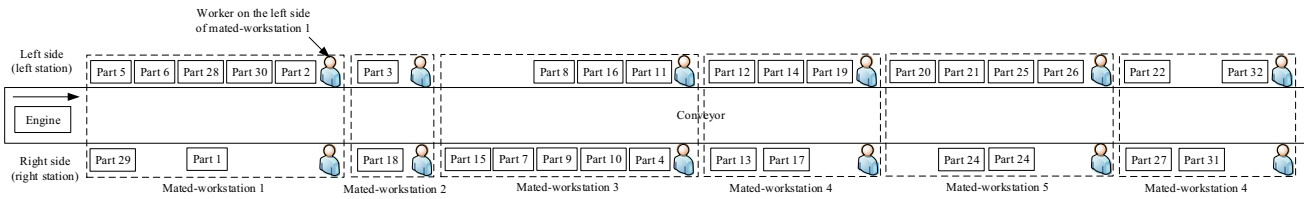
(a)



(b)



(c)



**Fig. 15** The assignment result of the selected disassembly schemes

in Pareto optimal set,  $\omega_i$  is a weighting index according to the importance value shown in Table 7.

$$J = \sum_{i=1}^4 \omega_i \times \frac{f_i - \min f_i}{\max f_i - \min f_i}. \quad (24)$$

By comparison and scoring, the judgment matrix is as follows:

$$\begin{bmatrix} 1 & 2 & 3 & 3 \\ 1/2 & 1 & 3 & 3 \\ 1/3 & 1/3 & 1 & 1 \\ 1/3 & 1/3 & 1 & 1 \end{bmatrix}.$$

Finally, the weighting value  $\omega_i$  can be obtained from the judgment matrix by  $\omega_i = \frac{1}{4} \times \sum_{j=1}^4 (a_{i,j} / \sum_{i=1}^4 a_{i,j})$ , and  $\omega_1 = 0.439$ ,  $\omega_2 = 0.313$ ,  $\omega_3 = 0.124$ , and  $\omega_4 = 0.124$ . Thus, the optimal solution in the Pareto front of IMOABC is the first type in Table 6, and the Gantt chart of the best solution (solution 1) is shown in Eq. (14), and Fig. 15 illustrates this task distribution result.

Through the above analysis, the proposed methodology can balance workload distribution, improve disassembly efficiency, reduce the number of workstations for cost reduction. For this case study, three kinds of solutions generated by IMOABC are provided for choosing. Compared with considering each objective lexicographically, Pareto non-dominated relationship can consider each goal, the results obtained by IMOABC is better than that of IMOABC. All solutions consider minimizing the number of mated-workstations, improving efficiency, rationally distributing workload, and simplifying the auxiliary operations. Also, decision-makers can choose a specific disassembly scheme based on their actual demands and the AHP method.

## 5 Conclusion and Future Work

The majority of published DLBP works are mainly based on a straight one-sided disassembly line, only one part can be removed at a time, and this linear production mode leads to

a long work time. This work studied DLBP based on a two-sided disassembly line for disassembly efficiency improvement and construction costs reduction. The main research contributions and novelties are as follows:

1. In order to describe the constraints among parts, a hybrid graph is used, and a connection matrix and a precedence matrix are constructed to express the relationship among parts in detail. On this basis, an improved disassembly sequence encoding method is customized for a two-sided disassembly line.
2. In response to the challenge in a two-sided disassembly line scheme planning, a decoding method is established for task assignment.
3. Aiming to reduce the number of mated-workstations, idle time, auxiliary indicator, and smooth index, a multi-objective two-sided disassembly line balancing model is established, and an improved multi-objective artificial bee colony algorithm which based on Pareto non-dominated relationship, crossover and mutation operators of GA is proposed for disassembly scheme optimization.

To verify the practicality and superiority of the proposed method and model, a case of an automotive engine is studied and compared with three typical algorithms. The results show that the proposed model and methodology can be applied to two-sided disassembly scheme planning and improve the scheme optimization performance of a two-sided disassembly line by the analysis in Sect. 4.1. Besides, the result indicates that the developed methodology can guide the disassembly of end-of-life products on a two-sided disassembly line.

Although the effectiveness of the proposed methodology has been verified, some limitations still exist. In this paper, the disassembly line is only designed for a single product. It is necessary to introduce a mixed-model for a two-sided disassembly line, which can handle various products by one line. Also, more uncertainties need to be considered, such as disassembly time might be changed, and some parts in the end-of-life sample product may be lost. Some objectives should be taken into consideration, such as profit and energy consumption.



**Acknowledgements** This work is financially supported by the National Natural Science Foundation of China under Grant No.51875156 and the National Key R&D Program of China under Grant No.2018YFC1902304.

**Funding** National Natural Science Foundation of China (No. 51875156), the National Key R&D Program of China (2018YFC1902304).

## Declarations

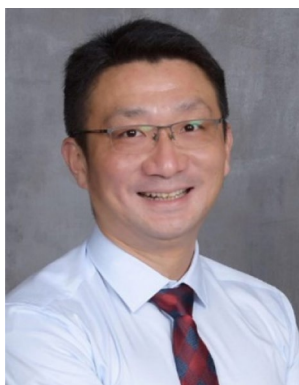
**Conflict of interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- Lee, C. M., Woo, W. S., & Roh, Y. H. (2017). Remanufacturing: Trends and Issues. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 4(1), 113–125.
- Kerin, M., & Pham, D. T. (2019). A review of emerging industry 4.0 technologies in remanufacturing. *Journal of Cleaner Production*, 237, 117805.
- Kalayci, C. B., & Gupta, S. M. (2014). A tabu search algorithm for balancing a sequence-dependent disassembly line. *Production Planning & Control*, 25(2), 149–160.
- Ozceylan, E., Kalayci, C. B., Gungor, A., & Gupta, S. M. (2019). Disassembly line balancing problem: A review of the state of the art and future directions. *International Journal of Production Research*, 57(15–16), 4805–4827.
- Gupta, S., & Gungor, A. (1999). Disassembly Line Balancing. In *Proceedings of the 1999 Annual Meeting of the Northeast Decision Sciences Institute*, 193–195.
- Battaia, O., & Dolgui, A. (2013). A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics*, 142(2), 259–277.
- Laili, Y., Li, Y., Fang, Y., Pham, D. T., & Zhang, L. (2020). Model review and algorithm comparison on multi-objective disassembly line balancing. *Journal of Manufacturing Systems*, 56, 484–500.
- Tian, G., Zhou, M., & Li, P. (2018). Disassembly sequence planning considering fuzzy component quality and varying operational cost. *IEEE Transactions on Automation Science and Engineering*, 15(2), 748–760.
- Ren, Y., Zhang, C., Zhao, F., Tian, G., Lin, W., Meng, L., et al. (2018). Disassembly line balancing problem using interdependent weights-based multi-criteria decision making and 2-Optimal algorithm. *Journal of Cleaner Production*, 174, 1475–1486.
- Ilgin, M. A. (2019). A DEMATEL-based disassembly line balancing heuristic. *Journal of Manufacturing Science and Engineering-Transactions of the Asme*, 141(2), 021002.
- Edis, E. B., Ilgin, M. A., & Edis, R. S. (2019). Disassembly line balancing with sequencing decisions: A mixed integer linear programming model and extensions. *Journal of Cleaner Production*, 238, 117826.
- Mete, S., Cil, Z. A., Celik, E., & Ozceylan, E. (2019). Supply-driven rebalancing of disassembly lines: A novel mathematical model approach. *Journal of Cleaner Production*, 213, 1157–1164.
- Kucukkoc, I. (2020). Balancing of two-sided disassembly lines: Problem definition, MILP model and genetic algorithm approach. *Computers & Operations Research*, 124, 105064.
- McGovern, S. M., & Gupta, S. M. (2007). A balancing method and genetic algorithm for disassembly line balancing. *European Journal of Operational Research*, 179(3), 692–708.
- Wang, S., Guo, X., & Liu, J. (2019). An efficient hybrid artificial bee colony algorithm for disassembly line balancing problem with sequence-dependent part removal times. *Engineering Optimization*, 51(11), 1920–1937.
- Kalayci, C. B., & Gupta, S. M. (2013). A particle swarm optimization algorithm with neighborhood-based mutation for sequence-dependent disassembly line balancing problem. *International Journal of Advanced Manufacturing Technology*, 69(1–4), 197–209.
- Kalayci, C. B., & Gupta, S. M. (2013). Ant colony optimization for sequence-dependent disassembly line balancing problem. *Journal of Manufacturing Technology Management*, 24(3), 413–427.
- Cil, Z. A., Mete, S., & Serin, F. (2020). Robotic disassembly line balancing problem: A mathematical model and ant colony optimization approach. *Applied Mathematical Modelling*, 86, 335–348.
- 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(3), 1089–1098.
- Ren, Y., 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(24), 7302–7316.
- Liu, J., Zhou, Z., Pham, D. T., Xu, W., Ji, C., & Liu, Q. (2020). Collaborative optimization of robotic disassembly sequence planning and robotic disassembly line balancing problem using improved discrete Bees algorithm in remanufacturing. *Robotics and Computer-Integrated Manufacturing*, 61, 01829.
- Yang, Y., Yuan, G., Zhuang, Q., & Tian, G. (2019). Multi-objective low-carbon disassembly line balancing for agricultural machinery using MDFOA and fuzzy AHP. *Journal of Cleaner Production*, 233, 1465–1474.
- Wang, K., Li, X., Gao, L., & Li, P. (2020). Modeling and balancing for green disassembly line using associated parts precedence graph and multi-objective genetic simulated annealing. *International Journal of Precision Engineering and Manufacturing-Green Technology*. <https://doi.org/10.1007/s40684-020-00259-7>
- Zhang, L., Zhao, X., Ke, Q., Dong, W., & Zhong, Y. (2021). Disassembly line balancing optimization method for high efficiency and low carbon emission. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 8(1), 233–247.
- Xia, X., Liu, W., Zhang, Z., Wang, L., Cao, J., & Liu, X. (2019). A balancing method of mixed-model disassembly line in random working environment. *Sustainability*, 11(8), 2304.
- Yang, W., & Cheng, W. (2020). Modelling and solving mixed-model two-sided assembly line balancing problem with sequence-dependent setup time. *International Journal of Production Research*, 58(21), 6638–6659.
- Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3), 459–471.
- Aslan, S., Badem, H., & Karaboga, D. (2019). Improved quick artificial bee colony (iqABC) algorithm for global optimization. *Soft Computing*, 23(24), 13161–13182.
- Kalayci, C. B., & Gupta, S. M. (2013). Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem. *Expert Systems with Applications*, 40(18), 7231–7241.
- Wang, K., Li, X., & Gao, L. (2019). A multi-objective discrete flower pollination algorithm for stochastic two-sided partial disassembly line balancing problem. *Computers & Industrial Engineering*, 130, 634–649.
- Zhang, X., Yu, G., Hu, Z., Pei, C., & Ma, G. (2014). Parallel disassembly sequence planning for complex products based on

- fuzzy-rough sets. *International Journal of Advanced Manufacturing Technology*, 72(1–4), 231–239.
32. Tian, G., Ren, Y., Feng, Y., Zhou, M., Zhang, H., & Tan, J. (2019). Modeling and planning for dual-objective selective disassembly using AND/OR graph and discrete artificial bee colony. *IEEE Transactions on Industrial Informatics*, 15(4), 2456–2468.
  33. Wang, Y., Li, F., Li, J., Chen, J., Jiang, F., & Wang, W. (2009). Hybrid graph disassembly model and sequence planning for product maintenance. In *International Technology and Innovation Conference 2006 (ITIC 2006)*, 515–519.
  34. 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.
  35. Adibi, M. A., & Shahrabi, J. (2014). A clustering-based modified variable neighborhood search algorithm for a dynamic job shop scheduling problem. *International Journal of Advanced Manufacturing Technology*, 70(9–12), 1955–1961.
  36. Wang, K., Li, X., & Gao, L. (2019). Modeling and optimization of multi-objective partial disassembly line balancing problem considering hazard and profit. *Journal of Cleaner Production*, 211, 115–133.
  37. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.
  38. Bader, J., & Zitzler, E. (2011). HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1), 45–76.
  39. Rostami, S., & Neri, F. (2017). A fast hypervolume driven selection mechanism for many-objective optimisation problems. *Swarm and Evolutionary Computation*, 34, 50–67.
  40. 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(2), 321–354.
  41. Sitorus, F., Cilliers, J. J., & Brito-Parada, P. R. (2019). Multi-criteria decision making for the choice problem in mining and mineral processing: Applications and trends. *Expert Systems with Applications*, 121, 393–417.
  42. Saaty, T. L. (1979). A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 15(3), 234–281.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Lei Zhang** is a Professor at the School of Mechanical Engineering, Hefei University of Technology (HFUT), Hefei, China. His research interests are product green manufacturing, green design theory and methods, computer-aided design, and product life cycle assessment.



**Yuanfeng Wu** received an M.S. degree from HFUT, Hefei, China, in 2021, during which he focused on remanufacturing system modeling and multi-objective optimization method. He is pursuing the Ph.D. degree in the School of Mechanical Engineering of HFUT. His current research interests include vibrations and waves in the continuous mechanical system, nonlinear dynamics, and intelligent optimization theory.



**Xikun Zhao** Xikun Zhao received the M.S. degree from HFUT, Hefei, China, in 2019. He is currently pursuing the Ph.D. degree in Mechanical Engineering College of Chongqing University, Chongqing, China. His research focuses on process modeling and intelligent optimization methods for sustainable manufacturing systems.



**Shiwen Pan** received his M.S. degree from HFUT, Hefei, China, in 2021. His primary research interests include green design and remanufacturing theory, data mining, and product life cycle assessment.



**Ziqi Li** is a Ph.D. student at the School of Mechanical Engineering of HFUT, Hefei, China. His research focuses on the green design of electromechanical products.



**Hong Bao** received the BE and Ph.D. degrees in mechanical engineering from the HFUT, Hefei, China, in 2005 and 2013, respectively. He is currently an associate professor of the School of Mechanical Engineering at HFUT. His current researches include low-carbon design and product life cycle assessment.



**Yongtin Tian** is a Ph.D. student at the School of Mechanical Engineering of HFUT. His main direction of scientific activity is modeling and improvement of manufacturing processes. His current research interests include green manufacturing, optimization of the recovery network model of the service terminal, and the disassembly design of large-scale electrical equipment.