Improved Fruit Fly Algorithm for Multi-Objective Disassembly Line Balancing Problem Considering Learning Effect

Haitao Zhang, Duc Truong Pham, and Qi Kang

***Abstract*—Disassembly lines break down scrapped products to recover valuable components for recycling and remanufacturing. Their productivity depends on several key factors: workstation operating costs, task priority, worker skill levels, and learning efficiency. This study considers the learning effect of disassembly and assembly workers. It pursues two objectives simultaneously: 1) maximizing the disassembly profit and 2) maximizing the learning outcomes of the workers. A bi-objective mixed integer programming model for the disassembly and assembly balancing problem is established to explore the optimal solution. A multi-objective fruit fly optimization algorithm is proposed, which can better handle multi-objective optimization problems by simulating the foraging behavior of fruit flies and finding a set of high-quality solutions that balance multiple objectives. The algorithm is compared with other multi-objective optimization algorithms. The experimental results show that the algorithm has outperform advantages.**

Keywords:Disassembly line balancing, multi-skilled workers, learning effect, Muti-objective fruit fly optimization algorithm, simulation

I. Introduction

Both recycling—recovering materials from discarded products—and reusing—extending product life by using items again—are effective strategies for conserving resources and reducing environmental impact [1]. The efficiency of the disassembly line plays a crucial part in achieving this goal, as it helps ensure the completeness of the product life cycle [2]. With automation integration, disassembly lines are characterized by high work efficiency [3]. However, this efficiency level does not guarantee maximum output from the disassembly line. Factors such as productivity, operating time, and disassembly order can significantly impact the output. Therefore, the disassembly line balancing problem (DLBP) is introduced in this context [4]. DLBP requires assigning disassembly tasks to an ordered sequence of workstations with a disassembly precedence relationship while optimizing the effectiveness of specific measures [5]. DLBP was first proposed by Gungor and Gupta [6], [7] in 2001 and then described as a multi-objective combinatorial problem. Laili *et al.* [8] classified the existing types of disassembly line problems, including the layout of disassembly lines and disassembly objectives. In addition, they summarized the modeling of disassembly lines, including some standard variables and disassembly constraints in the model. In recent years, research on DLBP has become more in-depth. For example, Qin *et al.* [9] established a parallel disassembly equilibrium model considering the employment of government welfare workers and verified the correctness of the proposed model through the exact solution results of CPLEX [10]. Guo *et al.* [11] established a mathematical model for the U-shaped layout disassembly line balance problem after considering many factors such as the disassembly performance of the human body, the fatigue level of the workers at the workstation, the disassembly profit, and the task precedence relationship. A multi-objective evolutionary algorithm was used to solve the proposed problem.

In the manufacturing industry, distribution planning is one of the key factors influencing the production efficiency of the disassembly line. DLBP has long been a hot research topic, yet only some studies explore the impact of multi-skilled workers on this issue. A disassembly line is a system comprising multiple workstations where products are sequentially processed by workers [12]. Flexibility in production systems is crucial in today's world. Multi-skilled workers enhance this flexibility by serving as a buffer to various demands [13]. Typically, these workers have several skills at different proficiency levels [14]. These proficiency levels affect the time required to process products [15]. Turan *et al.*[16] studied the multi-skilled labour planning problem, improved the efficiency of the maintenance network by optimizing the labour capacity of the repair shop and achieving labour heterogeneity through cross-training, and finally established a resilient maintenance service network for high-value assets. Guo *et al.* [17] studied the balancing problem of multi-product parallel disassembly lines, considering multi-skilled workers, and established a multi-objective mathematical model of parallel disassembly lines. Wang *et al.* [18] combined linear disassembly lines and U-shaped disassembly lines and considered multi-skilled workers, and established a hybrid disassembly line balancing problem with profit and carbon emissions as the goals. Efficiently allocating multi-skilled workers enhances efficiency and reduces costs. Therefore, integrating multi-skilled workers into the DLBP

Manuscript received June 28, 2025; revised July 14 and July 26, 2025; accepted July 27, 2025. This article was recommended for publication by Editor Shujin Qin upon evaluation of the reviewers' comments.

H. Zhang is with the College of Artificial Intelligence and Software, Liaoning Petrochemical University, Fushun 113001, China (e-mail: ZhangHaitao4502@163.com).

D. Pham is with the Department of Mechanical Engineering, University of Birmingham, Birmingham, B15 2TT, UK (e-mail: d.t.pham@bham.ac.uk).

Q. Kang is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, 07104, USA (e-mail: qk22@njit.edu).

Corresponding Author: Qi Kang

aims to increase system resilience and efficiency by optimizing the assignment of tasks and workers to workstations.

During the disassembly line production process, workers' learning efficient significantly impacts the efficiency of the entire disassembly line. From previous research on worker learning effects, Salameh used human learning effects in a production inventory model [19]. Erfan *et al.* [20] proposed a bi-objective mixed integer linear programming model considering the learning effect for allocating and scheduling disaster rescue units [21]. Mor B *et al.* [22] applied learning effects to flow shop scheduling problems. Azzouz A *et al.* [23]proposed and discussed a classification scheme for different scheduling models under learning effects. Currently, there are limited studies in this area. Workers often exhibit significant efficiency advantages when performing skilled tasks. They can improve their efficiency through accumulated experience during tasks, reducing the required working time and minimizing losses in disassembly activities. By considering the learning abilities of multi-skilled workers and the characteristics of the tasks, whole operations can be finely tuned to enhance the efficiency and quality of the entire disassembly line.

Based on the above reasons, this work studies DLBP considering worker learning effects, using an improved multi-objective fruit fly optimization algorithm (MOFOA) to maximize profits and enhance workers' learning outcomes. In recent years, the advancement of swarm intelligence optimization algorithms has led to their growing application in DLBP. These algorithms mimic various social animal behaviours observed in different groups, leveraging the interactions and collaborations between individual members to optimize performance. Guo *et al.* applied the randomized hybrid discrete grey wolf optimizer to the multi-objective decomposition sorting, and DLBP [24]. Luan *et al.* [25] used the improved whale algorithm to solve the flexible job shop scheduling problem. Fu *et al.* [26] et al. developed a multi-objective discrete fruit fly optimization algorithm combined with a stochastic simulation method to solve a stochastic multi-objective integrated disassembly-reprocessing-reassembly scheduling problem. This work applies a Pareto algorithm and can obtain a set of Pareto solutions [27]. Compared with the improved fruit fly optimization algorithms in existing literature, the proposed MOFOA also has some notable features. For example, compared with the single-objective fruit fly optimization algorithm in Qin *et al.* [28], the proposed algorithm introduces two main improvements: (1) This algorithm enables fruit flies to pursue multiple optimization goals simultaneously; (2) It use the Pareto solution set in each iteration as the position with the highest fitness. During the olfactory search process, each fruit fly individual in the population begins a random search from a random solution in the current generation's Pareto solution set.

Compared with the existing studies, this work makes the following two contributions.

1) This study considers the impact of learning effects on disassembly efficiency in practical factory worker training on the traditional disassembly line model. It solves the problem of task allocation and disassembly line balance for multi-skilled workers by quantifying and tracking the development of workers' experience and skills. Building on this foundation, we develop a multi-objective mixed integer programming model to maximize post-task profits and enhance the collective skill levels of the workforce.

2) An improved multi-objective fruit fly optimization algorithm is proposed. The algorithm uses a three-stage encoding square to represent the solution and designs an enhanced visual search method, taking the Pareto solution set in each iteration as the position with the highest fitness. Products of different specifications are combined into different multi-product cases for experiments and compared with other intelligent optimization algorithms. The results verify that the MOFOA algorithm performs better in solving the proposed DBLP problem than other multi-objective optimization algorithms.

The remaining sections are organized as follows: Section II describes the DLBP-MLW problem and establishes its mathematical model. Section III discusses the specific process of the improved multi-objective fruit fly optimization algorithm. Section IV details numerical experiments on several disassembly cases. Section V summarizes this paper.

II. Problem Statement and Formulation

A. Problem Description

DLBP-MLW, a sub-problem of DLBP, focuses on how the learning abilities of multi-skilled workers impact the efficiency of the disassembly line. This problem inherently incorporates the fundamental constraints of DLBP. In DLBP, tasks and workers are rationally assigned to workstations, adhering to task precedence and workstation cycle time constraints [29]. The primary objective of DLBP-MLW is to maximize the profitability of disassembly operations and enhance workers' overall skill levels upon task completion.

In DLBP, profit is calculated as the total value of disassembled parts minus disassembly costs. These costs encompass the workstation's start-up, fixed, and time-related costs incurred during the disassembly tasks.

In real-world scenarios, workers' efficiency is not constant and is typically influenced by the learning effect [30]. The learning effect refers to workers improving their skill level and efficiency by continuously accumulating experience during disassembly tasks [12]. Learning effects can be analyzed and interpreted in both short-term and long-term perspectives. From a long-term perspective, learning effects are usually represented by various learning curve models, including S-curve models, 2-parameter exponential models, and 3-parameter hyperbolic models, among others [31]. These models depict the relationship between experience accumulation and skill improvement, helping predict employee job performance. Although learning curves can accurately simulate the learning effect process, nonlinear models are often complex and challenging to solve. From a short-term perspective, the learning effect is viewed as a threshold phenomenon in which workers' productivity increases significantly once their experience accumulation reaches a certain threshold. This improvement can be modelled by introducing appropriate threshold parameters.

In this work, a simple model has been developed by establishing a systematic framework that associates the level of

worker experience with corresponding skill levels [32], [33]. By quantifying and tracking the progression of experience and skill acquisition, we can gain valuable insights into how workers' productivity and performance evolve. This model offers a practical and insightful approach to understanding the dynamics of learning and skill development in the context of worker performance. Each worker is assigned a different initial experience for each skill, with the relationship between skill level and skill experience represented as a piecewise function. The relationship between task execution time and work experience is shown in Fig. 1, in which E_{ps} represents the worker p 's experience of skill s, and T_{ir}^{l} represents the execution time of task i of product l at skill level r. To linearize the learning curve, we divide skills into different levels according to experience, and tasks require different times at different levels. Taking the execution time T_{im}^{l} under level m in the figure as an example, $E_{s(m)}^{lb}$ and $E_{s(m)}^{ub}$ represent the lower and upper bounds of the experience interval of skill s at level m, respectively. The determination of T_{im}^{l} value comes from the following formula:

$$(E_{s(m)}^{ub} - E_{s(m)}^{lb}) * T_{im}^{l} = \int_{E_{s(m)}^{lb}}^{E_{s(m)}^{ub}} f(e)de \tag{1}$$

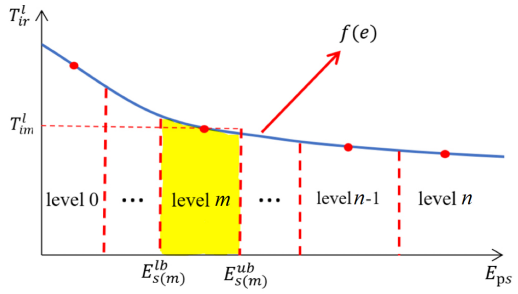

Fig. 1. Relationship between required time and skill level.

Assuming each task requires a specific skill, workers possess varying initial experience levels for each skill. Workers can gain corresponding skill experience through their learning ability while executing tasks. The amount of experience they gain is influenced by the duration of task execution. As workers continuously gain experience, their skill levels improve. By effectively assigning tasks to workers, the time cost of executing tasks can be reduced.

Regarding workstation and worker settings, each workstation is assigned only one worker. Each workstation has a cycle time limit, meaning that disassembly tasks must be completed within the specified cycle time at each workstation.

DLBP-MLW aims to obtain the maximum profit from product disassembly by considering both the disassembly time costs and the employment costs of workers. Ultimately, we must assign the most suitable workers to each workstation based on skill requirements.

B. Disassembly AND/OR Graph (DAOG)

Disassembly process can be rigorously specified with fromal methods[34][35]. In this paper, we illustrate the disassembly process using a disassembly AND/OR graph (DAOG), which provides detailed information about disassembly tasks, subassemblies, and the resulting disassembly parts. The AND/OR graph clearly shows the order of disassembly tasks and intuitively reveals the dependencies between subassemblies and parts. In the AND/OR graph, rectangles represent part numbers and information. Specifically, the part number is displayed as a number inside the angle brackets, and the details of the parts inside the part are marked outside the angle brackets. Each directed angle in the graph corresponds to a specific disassembly task, the task identifier is marked inside the angle, and the arrows intuitively depict the relationship between subassemblies. The starting point of the arrow points to the component that needs to be disassembled, and the direction of the arrow indicates the component obtained after performing the disassembly task. It should be noted that although there may be multiple directed angles around the rectangle, only one disassembly task can be performed on a component at a time, which means the disassembly is performed according to one directed angle each time. For example, after executing task 4 on component 2, we get components 4 and 8. A component can be torn down by executing different tasks. However, these tasks conflict with each other. When one of the tasks is executed, all other tasks are disabled. For example, component 5 can be torn down by either task 6 or task 7. These two tasks conflict with each other.

Fig. 2 shows an example of a treadmill consisting of 7 parts, while Fig. 3 describes the disassembly AND/OR graph of treadmill parts. DAOG requires different types of constraints, including conflict relationships, priorities of tasks, and relationships between tasks and sub-components. Fig. 3 shows that the disassembly process involves 18 sub-components, seven parts, and 17 disassembly tasks. During the disassembly process, task conflicts and precedence relationships must be satisfied. For example, after executing task 6 on part 5, we get parts 3 and 8. Parts can be disassembled by executing different tasks; however, these tasks may conflict. When a task is executed, all other tasks are disabled. For example, part 5 can be disassembled by task 6 or task 7, but these two tasks conflict.

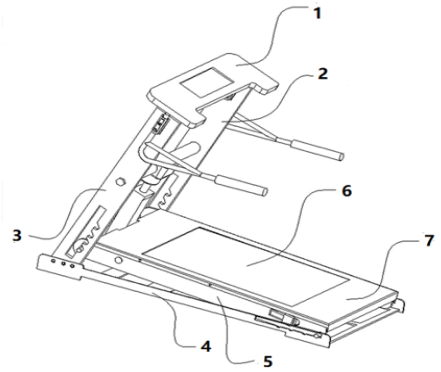

Fig. 2. Assembly diagram of a treadmill.

In order to correctly calculate the benefits of disassembly components, we define two matrices to describe the relationship between tasks and components and the relationship between tasks and skills.

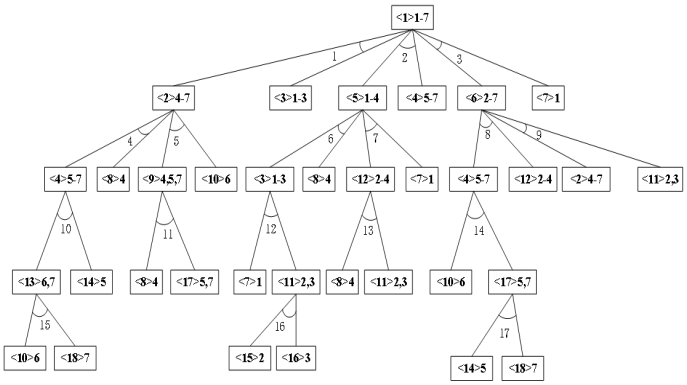

Fig. 3. The DAOG of a treadmill.

1) Subassembly-task association matrix

A subassembly-task association matrix $D = [d^l_{ji}]$ is constructed to define the relationships between subassemblies and tasks:

$$d^l_{ji} = \begin{cases} 1, \text{if subassembly } j \text{ is obtained by task } i \text{ in product } l; \\ -1, \text{if subassembly } j \text{ is disassembled by task } i \text{ in product } l; \\ 0, \text{otherwise} \end{cases}$$

2) Task-skill association matrix

A task-skill association matrix $B = [b^l_{is}]$ is used to describe the relationship between tasks and skills.

$$b^l_{is} = \begin{cases} 1, \text{ if disassembly task } i \text{ involves skill } s \text{ in product } l \\ 0, \text{ if disassembly task } i \text{ does not involve skill } s \text{ in product } l; \end{cases}$$

To build up a serial disassembly line, we assume that:

1) One worker is assigned to each workstation, and each worker has an initial experience.

2) Each disassembly task requires a disassembly skill. The higher the level of disassembly skill, the shorter the time required by the disassembly task.

3) Not all components in an EOL product need to be disassembled.

4) At least one disassembly task is assigned to each switched-on workstation.

C. *Notations*

Before we introduce the mathematical model, we list the notations to be used as follows:

Sets:

- $\mathbb{W}$ set of workstations.
- $\mathbb{S}$ set of skills.
- $\mathbb{P}$ set of workers.
- $\mathbb{L}$ set of products.
- $\mathbb{J}_l$ subassemblies set of product l.
- $\mathbb{I}_l$ task set of product l.
- $\mathbb{K}$ set of positions on each workstation.
- $\mathbb{R}$ set of skill levels.
- $\mathbb{I}^{lc}_i$ task set that conflict with task i of product l.
- $\mathbb{I}^{lp}_i$ preceding task set of task i of product l

Indexes:

- j subassembly index.
- i disassembly task index.
- l product index, $l = 1, 2, \ldots, L$, where L is the number of products.
- w workstation index, $w = 1, 2, \ldots, W$, where W is the number of workstations.
- s skill index, $s = 1, 2, \ldots, S$, where S is the number of skills.
- p worker index, $p = 1, 2, \ldots, P$, where P is the number of workers.
- k the order index of tasks performed at workstations, $k = 1, 2, \ldots, K$, where K is the number of tasks that can be performed at each workstation.
- r skill level index, $r = 1, 2, \ldots, R$, where R is the number of skill levels.

Parameters:

- T the cycle time of each workstation.
- T^l_{ir} the time required to perform task i of product l at skill level r
- C^l_{ir} the cost required to perform task i of product l at skill level r.
- E^0_{ps} the initial experience of worker p's disassembly skill s.
- E^{lb}_{sr} the lower bound of experience required for disassembly skill s at level r.
- C^W_w startup cost of workstation w.
- C_p cost of worker p.
- M a sufficiently large number.
- V^l_j benefits of subassembly j of product l.
- α_s learning effect coefficient of skill s.

Decision variables:

- x^l_{iwk} if task i of product l is executed at the k-th position on workstation w, $x^l_{iwk} = 1$; otherwise $x^l_{iwk} = 0$.
- z_{pw} if the p-th worker is assigned to the w-th workstation z_{pw}=1, otherwise $z_{pw} = 0$.
- u_w if the w-th workstation is started,$u_w = 1$; otherwise otherwise u_w= 0.
- y^l_{iwksr} if task i of product l is executed at the k-th position on workstation w and the level of skill s is r, then $y^{lp}_{iwksr} = 1$; otherwise $y^{lp}_{iwksr} = 0$.
- e_{wks} experience corresponding to the skill s of the worker at workstation w at position k.
- g_{psr} If the level of skill s of worker p is r after performing all tasks,$g_{psr} = 1$; otherwise $g_{psr} = 0$.
- f_{ps} The experience value of skill s for worker p after all tasks have been performed.

D. Mathematical Model

With the notations and decision variables listed above, the optimization problem of DLBP-MLW is formulated as follows and maximize (or minimize) the following objective function subject to the following constraints:

$$\max\left(\sum_{l\in\mathbb{L}}\sum_{j\in\mathbb{J}_l}\sum_{i\in\mathbb{I}_l}\sum_{w\in\mathbb{W}}\sum_{k\in\mathbb{K}} V_j^l d_{ji}^l x_{iwk}^l - \sum_{w\in\mathbb{W}} C_w^W u_w - \sum_{w\in\mathbb{W}}\sum_{p\in\mathbb{P}} C_p z_{pw} - \sum_{l\in\mathbb{L}}\sum_{k\in\mathbb{K}}\sum_{s\in\mathbb{S}}\sum_{w\in\mathbb{W}}\sum_{i\in\mathbb{I}_l}\sum_{r\in\mathbb{R}} C_{ir}^l y_{iwksr}^l\right) \tag{2}$$

$$\max\left(\sum_{s\in\mathbb{S}}\sum_{r\in\mathbb{R}}\sum_{p\in\mathbb{P}} r g_{psr}\right). \tag{3}$$

subject to:

$$\sum_{p\in\mathbb{P}} z_{pw} = u_w, \forall w \in \mathbb{W} \tag{4}$$

$$\sum_{w\in\mathbb{W}} z_{pw} \le 1, \forall p \in \mathbb{P} \tag{5}$$

$$f_{ps} = min(E_{ps}^0 + \sum_{l\in\mathbb{L}}\sum_{i\in\mathbb{I}_l}\sum_{k\in\mathbb{K}}\sum_{r\in\mathbb{R}}\sum_{w\in\mathbb{W}} \alpha_s T_{ir}^l y_{iwksr}^l E_{sr}^l b), \quad \forall p \in \mathbb{P}, \forall s \in \mathbb{S}. \tag{6}$$

$$\sum_{r\in\mathbb{R}} g_{psr} = 1, \forall p \in \mathbb{P}, \forall s \in \mathbb{S} \tag{7}$$

$$\sum_{l\in\mathbb{L}}\sum_{w\in\mathbb{W}}\sum_{k\in\mathbb{K}} x_{iwk}^l \le 1, \forall i \in \mathbb{I}_l \tag{8}$$

$$\sum_{l\in\mathbb{L}}\sum_{i\in\mathbb{I}_l} x_{iwk}^l \le 1, \forall w \in \mathbb{W}, \forall k \in \mathbb{K} \tag{9}$$

$$\sum_{l\in\mathbb{L}}\sum_{i\in\mathbb{I}_l} x_{iwk}^l \ge \sum_{l\in\mathbb{L}}\sum_{i\in\mathbb{I}_l} x_{iwk+1}^l, \forall w \in \mathbb{W}, \forall k \in \mathbb{K}\backslash\{K\} \tag{10}$$

$$\sum_{l\in\mathbb{L}}\sum_{w\in\mathbb{W}}\sum_{k\in\mathbb{K}} x_{iwk}^l + \sum_{l\in\mathbb{L}}\sum_{i'\in\mathbb{I}_i^{lc}}\sum_{w\in\mathbb{W}}\sum_{k\in\mathbb{K}} x_{i'wk}^l \le 1, \forall i \in \mathbb{I}_l \tag{11}$$

$$x_{iwk}^l \le \sum_{l\in\mathbb{L}}\sum_{i'\in\mathbb{I}_i^{lp}}\sum_{w'=1}^{w-1}\sum_{k'\in\mathbb{K}} x_{i'w'k'}^l + \sum_{l\in\mathbb{L}}\sum_{i'\in\mathbb{I}_i^{lp}}\sum_{k'=1}^{k-1} x_{iwk'}^l \quad \forall i \in \mathbb{I}_l, w \in \mathbb{W}, k \in \mathbb{K}\backslash\{1\} \tag{12}$$

$$\sum_{l\in\mathbb{L}}\sum_{i\in\mathbb{I}_l}\sum_{k\in\mathbb{K}} x_{iwk}^l \le M u_w, \forall w \in \mathbb{W} \tag{13}$$

$$\sum_{l\in\mathbb{L}}\sum_{i\in\mathbb{I}_l}\sum_{k\in\mathbb{K}} x_{iwk}^l \ge u_w, \forall w \in \mathbb{W} \tag{14}$$

$$e_{wks} = e_{wk-1s} + \sum_{l\in\mathbb{L}}\sum_{i\in\mathbb{I}_l}\sum_{r\in\mathbb{R}} \alpha_s T_{ir}^l y_{iwksr}^l \quad \forall w \in \mathbb{W}, s \in \mathbb{S}, \forall k \in \mathbb{K}\backslash\{1\} \tag{15}$$

$$e_{w1s} = \sum_{p\in\mathbb{P}} E_{ps}^0 z_{pw}, \forall w \in \mathbb{W}, \forall s \in \mathbb{S} \tag{16}$$

$$\sum_{l\in\mathbb{L}}\sum_{k\in\mathbb{K}}\sum_{s\in\mathbb{S}}\sum_{i\in\mathbb{I}_l}\sum_{r\in\mathbb{R}} T_{ir}^l y_{iwksr}^l \le T, \forall w \in \mathbb{W} \tag{17}$$

$$\sum_{r\in\mathbb{R}} y_{iwksr}^l = x_{iwk}^l b_{is}^l \quad \forall l \in \mathbb{L}, w \in \mathbb{W}, s \in \mathbb{S}, k \in \mathbb{K}, i \in \mathbb{I}_l \tag{18}$$

$$E_{sr}^{lb} - M\left(2 - y_{iwksr}^l - x_{iwk}^l\right) \le e_{wks} \le E_{sr+1}^{lb} + M\left(2 - y_{iwksr}^l - x_{iwk}^l\right) \quad \forall l \in \mathbb{L}, w \in \mathbb{W}, s \in \mathbb{S}, k \in \mathbb{K}, r \in \mathbb{R}, i \in \mathbb{I}_l \tag{19}$$

$$E_{sr}^{lb} - M\left(1 - g_{psr})\right) \le f_{ps} \le E_{sr+1}^{lb} + M\left(1 - g_{psr+1})\right) \quad \forall p \in \mathbb{P}, s \in \mathbb{S}, r \in \mathbb{R}\backslash\{1\} \tag{20}$$

$$x_{iwk}^l \in \{0,1\}, \forall i \in \mathbb{I}_l, w \in \mathbb{W}, k \in \mathbb{K}, l \in \mathbb{L} \tag{21}$$

$$z_{pw} \in \{0,1\}, \forall p \in \mathbb{P}, w \in \mathbb{W} \tag{22}$$

$$u_w \in \{0,1\}, \forall w \in \mathbb{W} \tag{23}$$

$$y_{iwksr}^l \in \{0,1\}, \forall i \in \mathbb{I}_l, w \in \mathbb{W}, k \in \mathbb{K}, s \in \mathbb{S}, r \in \mathbb{R}, l \in \mathbb{L} \tag{24}$$

$$e_{wks} \ge 0, \forall w \in \mathbb{W}, k \in \mathbb{K}, s \in \mathbb{S} \tag{25}$$

The objective function (2) aims to maximize profit, which is the total value of disassembled parts minus start-up workstations and time-related disassembly costs. The objective function (3) aims to maximize the maximum skill level of all workers after completing the task, i.e., to maximize the learning outcomes of workers. Constraint (4) specifies that a single worker is allocated to each activated workstation. Meanwhile, Constraint (5) mandates that each worker be assigned exclusively to one workstation. Constraint (6) calculates the worker experience after performing all tasks. Constraint (7) indicates that each worker's skill must correspond to a level. Constraint (8) specifies that any task is limited to a single execution per individual. Constraint (9) indicates that each spot on a workstation can host no more than one task. Constraint (10) ensures that lower-numbered workstations are used preferentially. This Constraint is used to reduce the solution space and improve model efficiency. Constraint (11) ensures that one of the conflicting tasks can be executed at most. Constraint (12) requires that before each task is executed, at least one of its immediate predecessor tasks must be completed. Constraint (13) states that tasks can only be assigned to started workstations. Constraint (14) means that each started workstation is assigned at least one task. Constraint (15) computes the worker's s-th skill experience before performing the task at the k-th position on the workstation, which is equal to the experience at the previous position plus the experience gained performing the task at the previous position. Constraint (16) states the first position of experience for each workstation, etc., on the initial experience of assigned workers. Constraint (17) means that the time to execute the task on the workstation must be less than the cycle time. Constraint (18) states that when task i is assigned to the k-th position on workstation w, only the corresponding skill has a unique level. Constraint (19) is a set of inequalities. When $y_{iwksr}^l = 1$, i.e., task i of the product l is performed

by the p-th worker at the k-th position of workstation w, and the level of skill s is r, according to the definition of the decision variable, it can be known that $x^l_{iwk} = 1$. In this case, Constraint (19) requires that the experience value of the p-th worker corresponding to skill level s should be within the corresponding interval. Constraint (20) is similar to (19). When $g_{psr} = 1$, i.e., when worker p has completed all tasks, the level of skill s is r. At this time, the corresponding skill experience of the worker is within the interval. Constraints (21) - (25) represent the value range of decision variables.

III. Proposed Algorithm

The Multi-Objective Fruit Fly Optimization Algorithm (MOFOA) is a biologically inspired optimization technique that models the foraging behaviour of fruit flies, specifically Drosophila, to address multi-objective optimization challenges. This algorithm is inspired by how fruit flies balance multiple goals, such as finding food sources while avoiding predators. MOFOA aims to concurrently identify the optimal solution set for multiple objectives by imitating this foraging behaviour.

A. Multi-Objective Fruit Fly Optimization Algorithm (MOFOA)

The multi-objective fruit fly optimization algorithm's core idea is to emulate fruit flies' foraging patterns, treating individual fruit flies as potential solutions, and the search space corresponds to the environment in which fruit flies food is searched. Specifically, the critical steps of the algorithm include:

1. Initialize the population: Randomly generate a group of fruit fly population individuals and distribute them in the problem's search space. Each population individual calculates its fitness value on multiple objective functions.

2. Update position: Following the rules of foraging behaviour in fruit flies, the location of each fruit fly is updated to explore and leverage the information within the search domain systematically.

3. Update fitness: Recalculate the fitness value of the updated fruit population individuals.

4. Determine the next-generation population position based on non-dominance rank order.

5. Termination condition judgment: Based on the initial termination condition, determine whether the conditions for stopping the algorithm are met. If satisfied, the algorithm ends; otherwise, return to step 3 to continue iteration.

The algorithm flow chart is shown in Fig. 4:

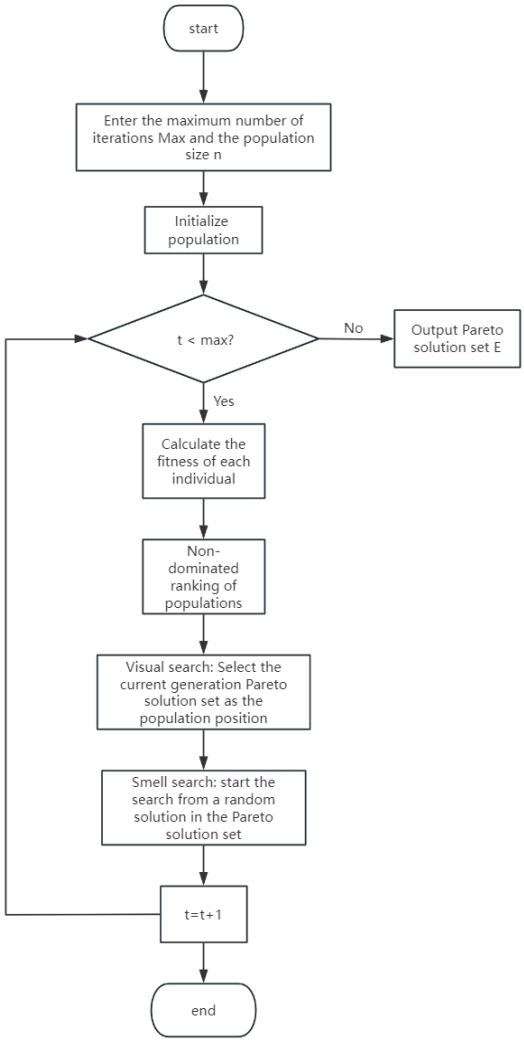

Fig. 4. Flowchart of multi-objective fruit fly optimization algorithm.

B. Encoding and Decoding

To solve the problem better, a three-stage encoding method $X\ (x_1, x_2, x_3)$ is adopted, corresponding to disassembly tasks, workstations, and workers, respectively. When generating new populations and individuals, integer permutation encoding is used to create disassembly task sequences of random lengths and make them feasible. First, a random task sequence is generated based on all task entities. The sequence is traversed to eliminate conflicting tasks according to the conflict matrix, and finally, the task order is adjusted to meet the precedence relationship. Next, disassembly tasks are randomly assigned to the workstations and it is ensured that the sum of the execution time of each task does not exceed the operation cycle of the workstation. Finally, a greedy strategy is used to preferentially select workers with high skill experience values to be assigned to workstations.

Fig. 5 illustrates this process using the DAOG of the treadmill in Fig. 3 as an example. First, we generate a random task sequence based on all the task entities in the graph. Then, we traverse the sequence and eliminate any conflicting tasks according to the conflict matrix. Finally, we adjust the order of tasks in the sequence to satisfy the precedence relationship between tasks. After the above steps, we get a feasible task sequence. Then, the disassembly tasks are randomly assigned to workstations in order. In the solution, we calculate the maximum execution time of each task to ensure that the total time of the disassembly task on a workstation is not greater than the operating cycle of the workstation. Finally, a greedy strategy is adopted to preferentially assign workers with high corresponding skill experience values to a workstation. The encoding structure is shown in Fig. 6. It

gives the structural composition of the optimal solution of Case 1 in the experimental part. The figure shows that the first-level integer sequence represents the disassembly task sequence. The second-level integer sequence represents the workstation number. For example, disassembly tasks 1, 4, and 12 are assigned to workstation 1, and tasks 16, 32, and 35 are assigned to workstation 2. The integer sequence in the third layer represents the number of workers. As shown in the figure, Worker Four is assigned to Workstation One, and Worker Six is assigned to Workstation 2.

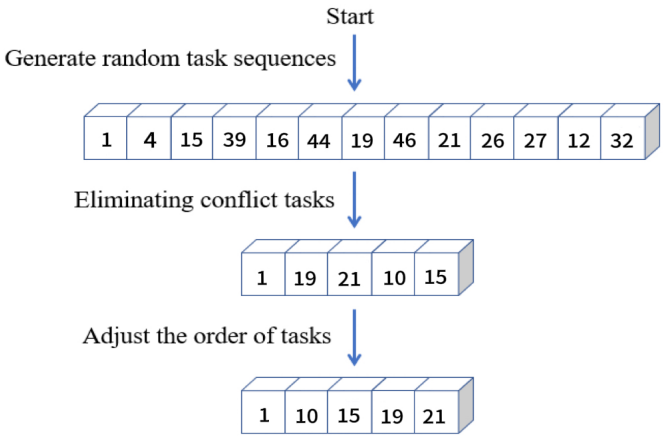

Fig. 5. Task Sequence Generation.

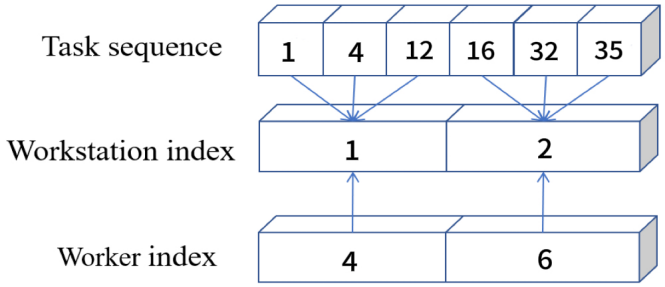

Fig. 6. Coding structure.

During the decoding process, the value of the disassembly component is first calculated according to the task sequence. The number of activated workstations and their costs are determined, the total cost of the workers is calculated, and finally, the time cost of executing the task is calculated. The objective function formula calculates the total price [28].

C. Smell Search and Visual Search

In the original FOA algorithm, the visual search process involves concentrating the fruit flies at the location of the fly with the highest fitness in the current population. In the Multi-Objective Fruit Fly Optimization Algorithm, the Pareto solution set in each iteration is considered the position with the highest fitness. During the olfactory search, each fruit fly randomly searches for a solution within the current generation's Pareto solution set. This random search is divided into four methods for the three-stage coding method: task sequence transformation, worker reallocation, and task reallocation. These methods are analogous to the search methods in the growth process of Section 4 [28]. Unlike the circular layout, where tasks can be randomly assigned to any workstation, the linear layout requires tasks to be redistributed according to the order of the workstations. Additionally, the allocated task execution time must be less than the cycle time of the workstation.

IV. Experimental Studies

A. Experimental Cases and Parameter Settings

This work studies the balancing problem of linear disassembly lines for various products. We selected four products with different specifications: a washing machine, a treadmill, a radio, and a hammer drill. These products were then combined into various multi-product cases for testing. Table I shows the details of these products. Table II shows size information for the combined case. According to the scale of the cases, Case 1 is a small-scale case, Case 2 is a medium-scale case, and Cases 3 and 4 are large-scale cases.

All experiments were conducted on Windows 11, using IntelliJ IDEA 2022 and the metal multi-objective optimization framework. We then combined and analyzed the results to draw comprehensive conclusions. In this chapter, we combined MOFOA with the electric potential energy evolutionary algorithm (ESPEA) [36], the non-dominated sorting genetic algorithm II (NSGA-II) [37], the volume metric-based non-dominated sorting steady-state multi-objective optimization algorithm (S-Metric Selection Evolutionary Multi-Objective Optimization Algorithm, SMS-EMOA) [38], the Pareto envelope-based selection algorithm II (PESA-II) [39] and the strength Pareto evolutionary algorithm II (SPEA-II) [40]. These algorithms were chosen as benchmarks because they are widely used, representative multi-objective evolutionary algorithms with different selection and diversity maintenance strategies, allowing a comprehensive evaluation of MOFOA's performance. The parameter settings are as follows: the population size and number of iterations are set to 100. To reduce the randomness of the intelligent optimization algorithm, ten experiments were conducted independently for each algorithm.

TABLE I Test product set.

Product	Num. of tasks	Num. of subassemblies	Num. of skills
Washing machine	13	15	3
Treadmill	17	18	3
Radio	30	29	3
Hammer drill	46	62	3

TABLE II Case information.

Case #	Product				Num. of tasks	Num. of skills
	Flash light	Washing machine	Radio	Hammer drill		
1	0	1	1	0	47	3
2	1	1	1	0	60	3
3	0	1	1	1	93	3
4	1	1	1	1	106	3

TABLE III Disassembly plan of MOFOA solution case 1

No.	Disassembly Sequence	f_1(Profit)	f_2(Level))
1	$(1, 19, 27, 12, 4, 21) \rightarrow 4$, $(32, 39, 44, 16, 46, 47, 26) \rightarrow 6$, $(10, 15) \rightarrow 3$	1519	32
2	$(1, 18, 5, 20, 21, 12, 11, 47, 22) \rightarrow 2$, $(22, 16, 23, 34, 17, 46, 26) \rightarrow 6$	1731	28
3	$(19, 1, 4, 47, 12, 27, 10, 21) \rightarrow 4$, $(22, 23, 34) \rightarrow 3$, $(16, 46, 15, 26) \rightarrow 6$	1610	31
4	$(2, 6, 14, 18, 20, 12, 21) \rightarrow 6$, $(47) \rightarrow 4$, $(22, 23, 34, 17, 16, 46, 26) \rightarrow 3$	1650	30
5	$(18, 1, 5, 20, 21, 11, 12, 22, 47) \rightarrow 2$, $(23, 34, 16, 46, 26, 17) \rightarrow 3$	1740	27
6	$(1, 19, 47, 12, 4, 27, 21) \rightarrow 12$, $(22, 23, 34, 46, 16, 26, 10, 15) \rightarrow 6$	1702	29

B. Analysis of Experimental Results

Table III shows the Pareto disassembly scheme for case 1 of the MOFOA solution, where f_1 and f_2 represent the disassembly profit and the worker skill level, respectively. The disassembly schemes in this table make trade-offs in different directions for the two optimization objectives of profit and level. Taking case 1 as an example, six different disassembly sequences are given. Among them, disassembly sequence 1 means that tasks 1, 19, 27, 12, 4, and 21 are assigned to workstation one and executed by worker 4, and tasks 32, 39, 44, 16, 46, 47, and 26 are assigned to workstation two and executed by worker 6. Tasks 10 and 15 are assigned to workstation three and executed by worker 3. The disassembly profit is 1519, and the maximum level is 32. Disassembly sequence 2 means that tasks 2, 18, 5, 20, 21, 12, 11, 47, and 22 are assigned to workstation one and executed by worker 2, and tasks 22, 16, 23, 34, 17, 46 and 26 are assigned to workstation two and executed by worker 6. The disassembly profit is 1731, and the maximum level is 28. It can be seen from the table that the MOFOA algorithm can find different optimal solutions for the same case, which can be selected according to the actual situation.

Table IV Comparison of multi-objective indicators and t-test results of different cases: comparison of multi-objective optimization indicators of each algorithm in different cases and t-test comparison of indicators. The indicators in the table include Hypervolume (HV), Epsilon indicator, Inverted Generational Distance plus (IGD+) and Relative Hypervolume (RHV). These indicators reflect the algorithm's diversity, convergence, and fit to the Pareto front. The "/" in the t-test in the table represents the MOFOA algorithm itself for comparison, "+" and "-" represent that the current evaluation index of MOFOA is better or worse than that of the algorithm, and "∼" indicates that the current index of MOFOA is not much different from that of the algorithm being compared. The data of each indicator in the table shows that MOFOA is generally better than other algorithms for comparison, whether small-scale or medium-to-large.

Intending to display the algorithm's performance more intuitively, Fig. 7 shows the Pareto front of each algorithm in each case. According to the figure, most Pareto solutions derived from MOFOA appear in the coordinate system's upper right corner. Compared with other algorithms, MOFOA has better optimization results on two objectives.

C. Comparison with Peer Algorithms

In the original FOA algorithm process, the visual search process concentrates the fruit flies on the location of the fruit flies with the highest fitness in the current population. In this study, we designed a new olfactory search process, which takes the Pareto solution set of each iteration as the position where fitness improves. Each fruit fly individual in the population starts the search from a random solution in the Pareto solution set of the current generation.

There is a certain degree of randomness in swarm intelligence algorithms. To verify the stability of MOFOA and its advantages over other intelligent algorithms, we also used the Non-Dominated Sorting Genetic Algorithm II (NSGA-II), the Pareto Envelope-Based Selection Algorithm II (PESA-II), the Intensity Pareto Evolutionary Algorithm (ESPEA), the Strength Pareto Evolutionary Algorithm II (SPEA-II), and the Stable Evolutionary Multi-Objective Algorithm (SMS-EMOA) based on S-metric and non-dominance to find solutions to the same situation. We conducted multiple independent experiments to compare the multi-objective optimization indicators of each algorithm in different cases. Through multi-objective indicator comparison and t-test results, we compared the Pareto fronts of each algorithm in different cases. Compared with similar algorithms, our analysis shows that MOFOA is simple and easy to implement, has fast convergence, and has strong global search capabilities. It can better avoid falling into local optima, adapt to many optimization problems, have relatively low computational complexity, and effectively handle conflicts between multiple objectives. It generates evenly distributed Pareto solution sets and has the advantage of solid robustness.

V. Discussion

This study extends the DLBP-MLW model by adding the objective of maximizing workers' skill levels. To address the resulting multi-objective problem, we develop an improved multi-objective fruit fly optimization algorithm (MOFOA) that uses the Pareto solution set in each iteration as the search center, thereby enhancing convergence and diversity. Experimental comparisons with ESPEA, NSGA-II, SMS-EMOA, PESA-II, and SPEA-II show that MOFOA achieves higher disassembly profits and faster skill accumulation. This improvement reflects better task allocation and more efficient utilization of

TABLE IV Comparison of multi-objective indicators and t-test results of different cases

Case	Algorithm	HV		Epsilon		IGD+		RHV	
		mean	t-test	mean	t-test	mean	t-test	mean	t-test
1	NSGAII	0.337925	+	0.319119	+	0.236402	+	0.404983	+
	MOFOA	0.431258	/	0.259811	/	0.153765	/	0.240642	/
	PESAII	0.344088	+	0.372453	+	0.258542	+	0.394131	+
	ESPEA	0.316541	+	0.369686	+	0.267936	+	0.442636	+
	SPEAII	0.391069	+	0.265283	~	0.194891	+	0.311406	+
	SMS-EMOA	0.381509	+	0.274088	~	0.19982	+	0.328239	+
2	NSGAII	0.352769	+	0.261991	~	0.225067	+	0.368233	+
	MOFOA	0.428416	/	0.252489	/	0.155244	/	0.232739	/
	PESAII	0.339909	+	0.325611	+	0.246014	+	0.391248	+
	ESPEA	0.334027	+	0.327059	+	0.247406	+	0.401783	+
	SPEAII	0.382353	+	0.252669	~	0.205773	+	0.315235	+
	SMS-EMOA	0.380543	+	0.225429	~	0.198161	+	0.318476	+
3	NSGAII	0.267199	+	0.419182	+	0.311683	+	0.552330	+
	MOFOA	0.461253	/	0.256969	/	0.135205	/	0.227209	/
	PESAII	0.248497	+	0.487212	+	0.347177	+	0.583664	+
	ESPEA	0.288203	+	0.456777	~	0.307393	+	0.51714	+
	SPEAII	0.283728	+	0.409974	+	0.297072	+	0.524638	+
	SMS-EMOA	0.312148	+	0.342647	+	0.260516	+	0.477022	+
4	NSGAII	0.360839	+	0.330419	+	0.245845	+	0.438622	+
	MOFOA	0.489744	/	0.264744	/	0.146427	/	0.238078	/
	PESAII	0.355973	+	0.387879	+	0.259028	+	0.446192	+
	ESPEA	0.406409	+	0.314219	+	0.222162	+	0.367724	+
	SPEAII	0.397494	+	0.323252	+	0.218766	+	0.381596	+
	SMS-EMOA	0.362966	+	0.312821	+	0.229095	+	0.435313	+

"~" indicates no significant difference between MOCPA and compared algorithms.

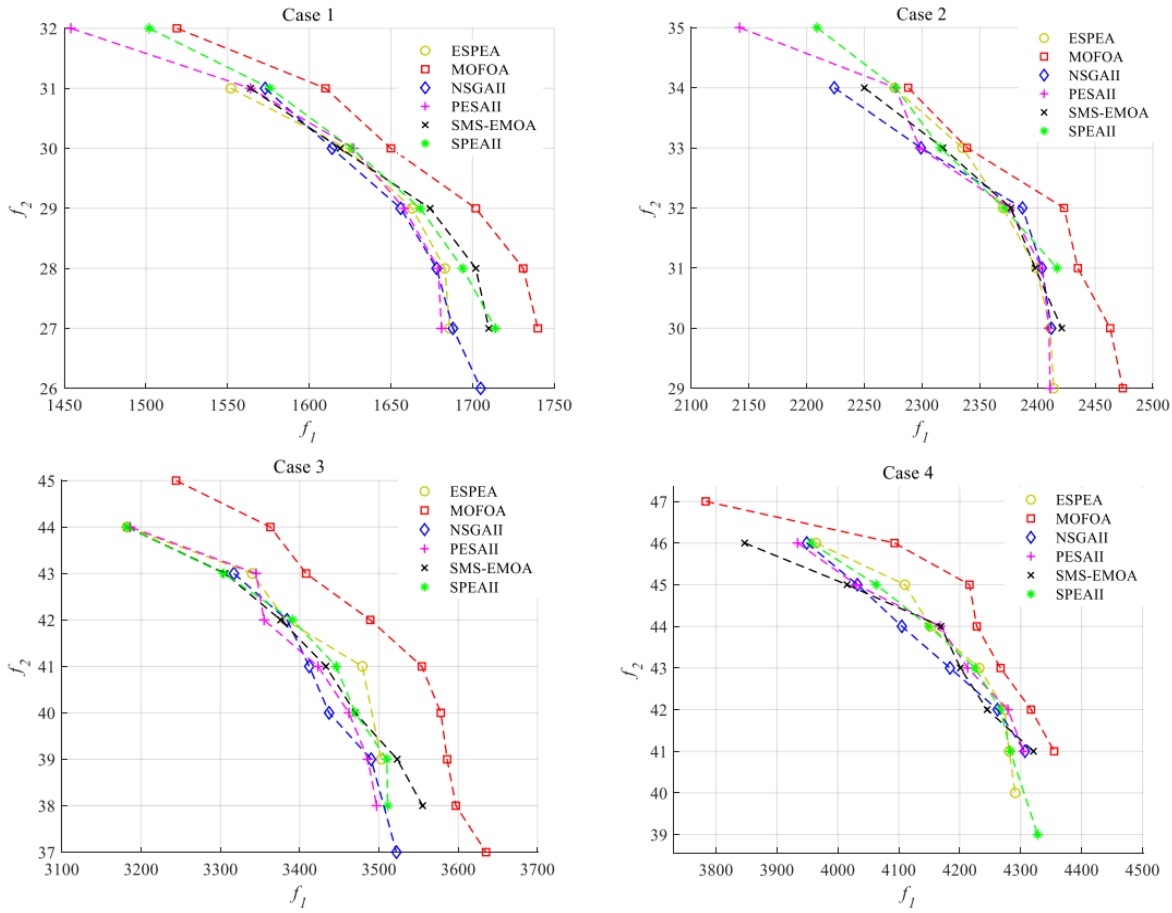

Fig. 7. Pareto front diagram

worker experience, ultimately enhancing production efficiency and product quality. However, the current model assumes fixed task requirements and worker configurations. Future research should address dynamic production environments and evaluate

scalability on larger problem instances to improve practical applicability.

This work proposes a multi-objective optimization model for disassembly line balancing, combining worker skill improvement with profit maximization. The enhanced MOFOA demonstrates superior performance compared to existing algorithms. Future research will incorporate dynamic objectives and explore more advanced optimization approaches—such as metaheuristic algorithms, hybrid algorithms, and hyper heuristic algorithms [41] to further enhance solution quality and adaptability [42][43].

REFERENCES

[1] X. Guo, M. Zhou, S. Liu, and L. Qi, "Multiresource-constrained selective disassembly with maximal profit and minimal energy consumption," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 2, pp. 804–816, 2020.

[2] Z. Zhao, "Research on influencing factors and balance issues of product disassembly lines," *Value Engineering 2010*, vol. 29, no. 26, pp. 256–257, 2010.

[3] Z. Zhang, X. Guo, M. Zhou, S. Liu, and L. Qi, "Multi-objective discrete grey wolf optimizer for solving stochastic multi-objective disassembly sequencing and line balancing problem," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 682–687.

[4] X. Wang, M. Zhou, Q. Zhao, S. Liu, X. Guo, and L. Qi, "A branch and price algorithm for crane assignment and scheduling in slab yard," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1122–1133, 2020.

[5] E. Özceylan and T. Paksoy, "Reverse supply chain optimisation with disassembly line balancing," *International Journal of Production Research*, vol. 51, no. 20, pp. 5985–6001, 2013.

[6] A. Gungor and S. M. Gupta, "Issues in environmentally conscious manufacturing and product recovery: a survey," *Computers & Industrial Engineering*, vol. 36, no. 4, pp. 811–853, 1999.

[7] ——, "A solution approach to the disassembly line balancing problem in the presence of task failures," *International journal of production research*, vol. 39, no. 7, pp. 1427–1467, 2001.

[8] Y. Laili, Y. Li, Y. Fang, D. T. Pham, and L. Zhang, "Model review and algorithm comparison on multi-objective disassembly line balancing," *Journal of Manufacturing Systems*, vol. 56, pp. 484–500, 2020.

[9] S. Qin, J. Li, J. Wang, X. Guo, S. Liu, and L. Qi, "A salp swarm algorithm for parallel disassembly line balancing considering workers with government benefits," *IEEE Transactions on Computational Social Systems*, 2023.

[10] L. Qi, M. Li, X. Guo, and W. Luan, "Multi-objective optimization for robotaxi dispatch with safety-carpooling mode in pandemic era," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 1, pp. 878–891, 2024.

[11] X. Guo, T. Wei, J. Wang, S. Liu, S. Qin, and L. Qi, "Multiobjective u-shaped disassembly line balancing problem considering human fatigue index and an efficient solution," *IEEE Transactions on Computational Social Systems*, 2022.

[12] R. L, M. L, F. C, F. Z, and C. C, "Eco-friendly multi-skilled worker assignment and assembly line balancing problem." *Computers Industrial Engineering*, vol. 106944, 2021.

[13] B. F. Nembhard D A, "Selection policies for a multifunctional workforce," *International Journal of Production Research*, vol. 52, no. 16, pp. 4785–4802, 2014.

[14] G. G. J. C. Bokhorst J A C, Slomp J, "On the who-rule in dual resource constrained (drc) manufacturing systems," *International Journal of Production Research*, vol. 42, no. 23, 2004.

[15] N. B. A, T. W, N. K. L, B. B, and W. R. C, "Worker assignment in cellular manufacturing considering technical and human skills," *International Journal of Production Research*, vol. 40, no. 6, pp. 1479–1492, 2002.

[16] H. H. Turan, F. Kosanoglu, and M. Atmis, "A multi-skilled workforce optimisation in maintenance logistics networks by multi-thread simulated annealing algorithms," *International Journal of Production Research*, vol. 59, no. 9, pp. 2624–2646, 2021.

[17] X. Guo, L. Zhou, Z. Zhang, L. Qi, J. Wang, S. Qin, and J. Cao, "Multi-objective optimization of multi-product parallel disassembly line balancing problem considering multi-skilled workers using a discrete chemical reaction optimization algorithm." *Computers, Materials & Continua*, vol. 80, no. 3, 2024.

[18] J. Wang, G. Xi, X. Guo, S. Qin, and H. Han, "Multi-objective advantage actor-critic algorithm for hybrid disassembly line balancing with multi-skilled workers," *Information*, vol. 15, no. 3, p. 168, 2024.

[19] J. M. Y. Salameh M K, Abdul-Malak M A U, "Mathematical modelling of the effect of human learning in the finite production inventory model," *Applied mathematical modelling*, vol. 17, no. 11, pp. 613–615, 1993.

[20] R.-B. M. Tirkolaee E B, Aydın N S, "A robust bi-objective mathematical model for disaster rescue units allocation and scheduling with learning effect," *Computers Industrial Engineering*, vol. 149, no. 106790, 2020.

[21] S. L. B. Azzouz A, Ennigrou M, "Scheduling problems under learning effects: classification and cartography," *International Journal of Production Research*, vol. 56, pp. 1642–1661, 2018.

[22] S. D. Mor B, Mosheiov G, "Flowshop scheduling with learning effect and job rejection," *Journal of Scheduling*, vol. 23, no. 6, pp. 631–641, 2020.

[23] B. S. L. Azzouz A, Ennigrou M, "Scheduling problems under learning effects: classification and cartography," *International Journal of Production Research*, vol. 56, no. 4, pp. 1642–1661, 2018.

[24] X. Guo, Z. Zhang, L. Qi, S. Liu, Y. Tang, and Z. Zhao, "Stochastic hybrid discrete grey wolf optimizer for multi-objective disassembly sequencing and line balancing planning in disassembling multiple products," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1744–1756, 2021.

[25] F. Luan, Z. Cai, S. Wu, T. Jiang, F. Li, and J. Yang, "Improved whale algorithm for solving the flexible job shop scheduling problem," *Mathematics*, vol. 7, no. 5, 2019.

[26] Y. Fu, M. Zhou, X. Guo, and L. Qi, "Stochastic multi-objective integrated disassembly-reprocessing-reassembly scheduling via fruit fly optimization algorithm," *Journal of Cleaner Production*, vol. 278, p. 123364, 2021.

[27] Y. Hou, N. Wu, M. Zhou, and Z. Li, "Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm," *IEEE T. Syst. Man. Cy-S.*, vol. 47, no. 3, pp. 517–530, 2017.

[28] S. J. Qin, J. P. Wang, J. C. Wang, S. X. Liu, X. W. Guo, L. Qi, and Z. Y. Zhao, "A multi-objective distribution-free model and method for stochastic disassembly line balancing problem," *IEEE Transactions on Computational Social Systems*, 2024.

[29] X. Guo, Z. Zhang, S. Qi Liang, Liu, Y. Tang, and Z. Ziyan, "Stochastic hybrid discrete grey wolf optimizer for multi-objective disassembly sequencing and line balancing planning in disassembling multiple products," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1744–1756, 2021.

[30] J. Wang, X. Guo, M. Zhou, J. Wang, S. Qin, and L. Qi., "Discrete fruit fly optimization algorithm for disassembly line balancing problems by considering human worker's learning effect," *Australian New Zealand Control Conference (ANZCC)*, 2022.

[31] M. J. Anzanello and F. S. Fogliatto, "Learning curve models and applications: Literature review and research directions," *International Journal of Industrial Ergonomics*, vol. 41, no. 5, pp. 573–583, 2011.

[32] M. T. Cezik and P. L'Ecuyer, "Staffing multiskill call centers via linear programming and simulation," *Management Science*, vol. 54, no. 2, pp. 310–323, 2008.

[33] S. Qin, S. Liu, and H. Kuang, "Piecewise linear model for multiskilled workforce scheduling problems considering learning effect and project quality," *Mathematical Problems in Engineering*, vol. 2016, pp. 1–11, 2016.

[34] J. Wang, Y. Deng, and C. Jin, "Performance analysis of traffic control systems based upon stochastic timed petri net models," *International Journal of Software Engineering and Knowledge Engineering*, vol. 10, no. 06, pp. 735–757, 2000.

[35] J. Wang, "Charging information collection modeling and analysis of gprs networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 4, pp. 473–481, 2007.

[36] G. Mancini, M. Fusè, F. Lazzari, and V. Barone, "Fast exploration of potential energy surfaces with a joint venture of quantum chemistry, evolutionary algorithms and unsupervised learning," *Digital Discovery*, vol. 1, no. 6, pp. 790–805, 2022.

[37] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[38] Z. Yang, H. Wang, K. Yang, T. Back, and M. Emmerich, "Sms-emoa with multiple dynamic reference points," *IEEE*, pp. 282–288, 2016.

[39] M. Li, S. Yang, X. Liu, and K. Wang, "Ipesa-ii: Improved pareto envelope-based selection algorithm ii," *Springer*, pp. 143–155, 2013.
[40] T. L. Zitzler E, Laumanns M, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK report*, vol. 103, 2001.
[41] E. Singh and N. Pillay, "A study of ant-based pheromone spaces for generation constructive hyper-heuristics," *Swarm and Evolutionary Computation*, vol. 72, p. 101095, 2022.
[42] Z. Bi, X. Guo, J. Wang, S. Qin, and G. Liu, "Deep reinforcement learning for truck-drone delivery problem," *Drones*, vol. 7, no. 7, 2023.
[43] K. Zhang, W. Lou, J. Wang, R. Zhou, X. Guo, Y. Xiao, C. Shi, and Z. Zhao, "Pa-detr: End-to-end visually indistinguishable bolt defects detection method based on transmission line knowledge reasoning," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–14, 2023.

HaiTao Zhang received her B.S. degree in Computer Science and Technology from Linyi University, China in 2022. Currently, she is a graduate student at the School of Artificial Intelligence and Software, Liaoning Shihua University, Fushun, China. Her research interests are disassembly line balancing problems and intelligent optimization algorithms.

Duc Truong Pham is a leading researcher in mechanical, manufacturing, computer, and systems engineering, with over 600 technical papers, 17 books, and supervision of 100+ PhD theses. He has secured £30M+ in research funding and collaborated with major companies to apply his work in industry. He has delivered 50+ keynote talks worldwide and held visiting positions at top institutions in France, China, New Zealand, and Saudi Arabia. His accolades include multiple IMechE prizes, a Lifetime Achievement Award (2016), and a Distinguished International Academic Contribution Award (2017). He is a Fellow of several prestigious engineering societies and was appointed OBE in 2003.

Qi Kang is a Ph.D candidate in Electrical and Computer Engineering department of New Jersey Institute of Technology. His research focuses the neural modulation with transcranial electrical stimulation and focused ultrasound stimulation. Prior to arriving at NJIT, he holds a Master of Science degree in Electrical and Computer Engineering from New York Institute of Technology in Old Westbury.