Human-Robot Collaborative Disassembly Profit Maximization via Improved Grey Wolf Optimizer

Zhiwei Zhang, Shaokang Dai, Chong Li, Weitian Wang, Jesse Parron, and Emilio Herrera

***Abstract*—With the continuous advancement of modern technologies, a growing number of new products feature increasingly complex designs. The complexity of these products poses significant challenges to the planning and execution of their disassembly processes. This work proposes a human-robot collaborative disassembly solution to address such challenges. A circular disassembly line layout enables the cyclic assignment of tasks between human operators and robots. This layout overcomes the time and spatial limitations inherent in traditional disassembly lines. A human-robot collaborative circular disassembly line balancing problem is formulated, along with its integer mathematical programming model aimed at maximizing disassembly profit. The competency of the model is verified by applying the commercially available software CPLEX to small-scale product examples. An improved grey wolf optimizer is proposed to solve large instances that CPLEX fails to address. Its experimental results are compared with those of several representative intelligent optimizers. The results indicate that it surpasses existing methods, highlighting its potential as a promising solution for industrial applications.**

***Key Words*—Human-robot collaboration, Circular disassembly line, Mixed-integer programming, Grey wolf optimizer.**

I. Introduction

THE continuous development of modern society, particularly in the economic and technological sectors, has led to an increasing rate of obsolescence of industrial machines, equipment, and electronic products [1, 2, 3, 4]. Disposing of end-of-life or discarded products places immense pressure on resources and the environment. Improper disposal results in the squandering of valuable resources and contributes to environmental pollution [5]. Upon reaching the end of its operational lifespan, industrial machinery is often landfilled or incinerated. Such disposal contributes to the accumulation of waste and releases hazardous substances, such as lead and tungsten, into the soil, water bodies, and the atmosphere. These releases have severe detrimental effects on human health and ecosystems. Disassembly is a sustainable production method that transforms end-of-life or discarded products into resources and reusable components, promoting material recovery and recycling [6, 7]. This method reduces the manufacturing costs of new products and fosters the development of related remanufacturing industries [8]. It establishes a complete remanufacturing industry chain, providing a new driving force for economic growth.

Disassembly is the process of dismantling used products into parts and materials [9]. Remanufacturing involves reprocessing disassembled parts to create new products. Therefore, disassembly is the prerequisite and foundation of remanufacturing, enabling its execution. The disassembly line balancing problem (DLBP) involves the optimal assignment of disassembly tasks to workstations while satisfying various constraints [10, 11, 12, 13]. Currently, four traditional types of disassembly line layouts exist: linear, U-shaped, two-sided, and parallel layouts [14, 15, 16, 17]. Each layout has its characteristics, but all are challenging to balance in task allocation, a problem also present in production systems. To address this problem, scholars have proposed a loop-closed layout, and research on this layout has continued uninterrupted for the past 20 years [18, 19, 20].

In this paper, inspired by the circular layout in flexible production systems [21], we propose a new type of disassembly line: the circular layout. These lines feature workstations arranged in a circular pattern. Unlike other disassembly line models, they facilitate the cyclic distribution of tasks based on their characteristics. Their layout and distribution scheme effectively overcome the space constraints of other disassembly lines while enhancing the flexibility of task allocation to workstations. With the advancement of artificial intelligence technology, robots have rapidly developed and are now widely used in industrial practice [22, 23, 24, 25]. Traditionally, disassembly work has been primarily performed manually by humans. The introduction of robots has partially automated the disassembly process. However, robots cannot fully replace humans due to the increased complexity involved in product disassembly. Although robots offer advantages in disassembly efficiency, speed, and safety, their ability to handle uncertainties and adapt to complex product structures remains limited [26, 27].

In contrast, humans possess unparalleled flexibility, enabling them to manage unexpected situations and perform delicate disassembly tasks more effectively than robots. However, when human operators disassemble dangerous subassemblies, insecurity may inevitably arise. Therefore, cooperation between humans and robots can complement their strengths [28, 29, 30] , resulting in a human-robot collaborative team that outperforms systems consisting solely of human operators or robots. Lee *et al.* [31] propose a human-robot collaboration-based disassembly task allocation and planning method that

Manuscript received June 25, 2025; revised July 10, 2025: accepted July 18, 2025. This article was recommended for publication by Associate Editor Shujin Qin upon evaluation of the reviewers' comments.

Z. Zhang, S. Dai, and C. Li are with the College of Artificial Intelligence and Software, Liaoning Petrochemical University, Fushun 113001, China (e-mail: 18340312375@163.com, 15751530086@163.com, 1902427447@qq.com).

W. Wang, J Parron, and E. Herrera are with the School of Computing, Montclair State University, Montclair, NJ 07043, USA (e-mail: wangw@montclair.edu, parronj@montclair.edu, herrerae7@montclair.edu).

Corresponding Author: Weitian Wang

considers limited resources and factors such as human safety. Based on human fatigue levels, Li *et al.* [32] apply a human-robot collaboration strategy and use the discrete bee algorithm to minimize disassembly time. Wu *et al.* [33] propose a novel human-robot collaborative disassembly line balancing problem. A mixed-integer programming model is designed to minimize the number of workstations, smoothing exponents, and various costs. A hybrid local search genetic algorithm is developed and applied to solve the disassembly problem for batteries. Tsarouchi *et al.* [34] apply human-robot collaboration to a hybrid assembly cell, implement a decision algorithm based on the evaluation of multiple criteria, and present the case of human-robot assembly of a hydraulic pump. Although many studies on human-robot collaboration operations exist, few studies address human-robot collaboration in circular disassembly lines. Our work addresses this gap.

We leverage the characteristics of a circular disassembly line to efficiently assign tasks to workstations, thereby enhancing the efficiency and quality of the entire disassembly process. A robot performs tedious and hazardous disassembly tasks, while a human executes delicate ones. Certain common tasks can be performed by either a human or a robot. This approach achieves the objective of human-robot collaborative disassembly, reduces costs, and ensures human safety [35, 36]. A mathematical planning model is developed for the task assignment problem. The model aims to maximize disassembly profit [37]. The modeling process comprehensively considers conflict relationships, precedence relationships, task allocation, and other relevant factors.

Swarm intelligence optimizers have been widely applied in DLBP research. Ding *et al.* [38] propose a novel multi-objective ant colony algorithm for solving multi-objective optimization problems. Guo *et al.* [39] propose an improved multi-objective shuffled frog leaping algorithm and apply it to human-robot collaborative disassembly line balancing. Wang *et al.* [17] apply a simulated annealing algorithm to solve a parallel DLBP. Ren *et al.* [40] propose a gravitational search-based algorithm for solving a profit-oriented parallel DLBP. Kalayci *et al.* [41] introduce an artificial bee colony algorithm for tackling a multi-objective serially correlated DLBP. Recent studies have also focused on applying specific algorithms in disassembly scheduling [42, 43, 44, 45, 46, 47]. The Grey Wolf Optimizer (GWO) is an intelligent global optimization algorithm inspired by the social behavior of grey wolves in nature [48, 49, 50, 51]. It offers fast convergence and strong global search capabilities and can be adapted to solve various problems by adjusting algorithm parameters. GWO has been extensively applied across multiple domains, including data mining [52], image processing [53], neural network optimization [54, 55], and scheduling problems [56]. In solving the CCDP described in this work, the hunting process of a grey wolf is simulated. The optimal solution is obtained by continuously adjusting the positions of the grey wolves, effectively solving the optimization problem. The contributions of this work are two-fold:

1) A human-robot collaboration approach is proposed to enhance disassembly efficiency. Given that load balancing among workstations facilitates the equal distribution of tasks between humans and robots, this paper adopts a circular disassembly line layout. This enlarges workstation utilization and improves the balance of human-robot collaboration. In this paper, a mathematical model is built to address the problem described above.
2) An improved grey wolf optimizer (IGWO) is developed. A three-stage encoding scheme is created for the assignment relationship among tasks, workstations, and operators. The hunting process of grey wolves is simulated through two behaviors: searching and attacking. This process enhances the global search capability, increases population diversity, and improves convergence performance.

In this work, CPLEX, a commercially available exact solver, is applied to verify the mathematical model's competency. The proposed solution is compared with those of several representative optimizers. Experimental results demonstrate that IGWO outperforms its peers in solving the problem.

The paper is structured into five main sections. Section II introduces the problem and its mathematical model. Section III describes the algorithms. Section IV outlines the experiments performed. Finally, Section V concludes this study and discusses our future work.

II. Problem Statement and Formulation

A. Problem Description

This work addresses a circular disassembly line layout with human-robot collaborative disassembly patterns. Fig. 1 illustrates the layout. A robot and a human workstation are positioned on the circular disassembly line to execute their assigned tasks. This design cyclically allocates tasks with the same attributes to the same workstation, thereby maximizing overall operational efficiency and balancing workloads among the disassembly workstations.

For effective disassembly, tasks are classified into three categories: dangerous, delicate, and common, based on the characteristics of the parts. Robots handle the disassembly of hazardous subassemblies, such as high-pressure or explosive components, thereby reducing the risk of accidents for human operators. Humans perform delicate disassembly tasks, such as handling computer chips and precision instruments, ensuring high precision and accuracy. For common tasks, either a human or robot can be used, and the decision is made based on the specific situation. In recent years, the human-robot collaboration model has seen significant adoption. Effective coordination between humans and robots benefits the system by improving efficiency, ensuring safety, and reducing disassembly costs.

For ease of modeling and use, we represent the relationships among tasks and the relationships between tasks and subassemblies as matrices.

- A precedence matrix $F = [f_{i,i'}]$. It describes the precedence relationship between two tasks, where i and i' represent two different disassembly tasks of a product, respectively.

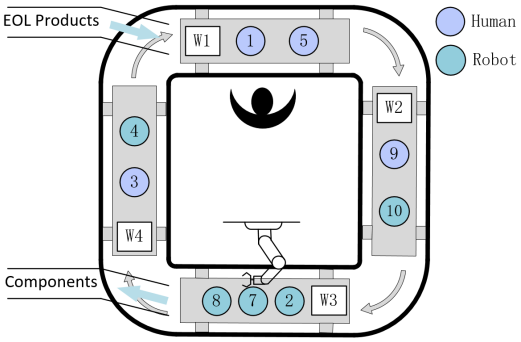


Fig. 1. Human-robot collaborative circular disassembly line.

$$f_{i,i'} = \begin{cases} 1, \text{if task } i \text{ is immediately executed before task } i' \\ 0, \text{otherwise} \end{cases}$$

- A conflict matrix $G = [g_{i,i'}]$. It describes the conflict relationship between two tasks.

$$g_{i,i'} = \begin{cases} 1, \text{if task } i \text{ is executed before task } i' \\ \text{-1, if task } i \text{ and task } i' \text{ conflict} \\ 0, \text{otherwise} \end{cases}$$

- A disassembly matrix $D = [d_{j,i}]$. It describes the disassembly relationship between subassembly j and task i.

$$d_{j,i} = \begin{cases} 1, \text{if subassembly } j \text{ is obtained through task } i \\ \text{-1, if subassembly } j \text{ is disassembled by task } i \\ 0, \text{otherwise} \end{cases}$$

In this study, we make the following assumptions:

1) The conflict relation G and the precedence relation F among tasks are known.
2) A workstation turned on is assigned at least one task for disassembly.
3) The disassembly cost of humans and robots for each task performed is known.
4) The benefit of each task execution is known.
5) Delicate and dangerous disassembly tasks are identified and known.

B. Symbol Definition and Mathematical Model

- c_i^R: cost for robot in executing disassembly task i.
- c_i^H: cost for human in executing disassembly task i.
- c_w^R: the start-up cost of robot workstation w.
- c_w^H: the start-up cost of human workstation w.
- d_{ji}: the disassembly relationship between subassembly j and task i.
- I: number of tasks.
- $\mathbb{I}$: set of task $\mathbb{I} = \{1, 2..., I\}$.
- $\mathbb{I}_i^G$: set of conflict relationships for task i.
- $\mathbb{I}_i^F$: set of precedence relationships for task i.
- $\mathbb{I}^R$: set of dangerous tasks to be disassembled by a robot.
- $\mathbb{I}^H$: set of delicate tasks to be disassembled by a human.
- J: number of subassembly.
- $\mathbb{J}$: set of subassembies $\mathbb{J} = \{1, 2..., J\}$.
- K: Number of task execution sequences.
- $\mathbb{K}$: set of task positions on each workstation $\mathbb{K} = \{1, 2..., K\}$.
- p_j: the value of subassembly j.
- t_i^R: time for robot in executing disassembly task i.
- t_i^H: time for human in executing disassembly task i.
- W: number of workstations.
- $\mathbb{W}$: set of workstation $\mathbb{W} = \{1, 2..., W\}$.

1) Decision Variables:

$$h_i = \begin{cases} 1, \text{ if task } i \text{ must be disassembled by a human} \\ 0, \text{otherwise} \end{cases}$$

$$r_i = \begin{cases} 1, \text{ if task } i \text{ must be disassembled by a robot} \\ 0, \text{otherwise} \end{cases}$$

$$x_{i,k,w} = \begin{cases} 1, \text{if task } i \text{ is executed at the } k\text{th position} \\ \text{on workstation } w \\ 0, \text{otherwise} \end{cases}$$

The "otherwise" in h_i and r_i has two meanings. When r_i = 0, task i may be disassembled by a human or not be done, implying selective disassembly. When h_i = 0, task i may be disassembled by a robot or not be done.

$$s_w = \begin{cases} 1, \text{if the workstation w is turned on} \\ 0, \text{otherwise} \end{cases}$$

$$q_w^R = \begin{cases} 1, \text{if the robot disassembly workstation } w \text{ is turned on} \\ 0, \text{otherwise} \end{cases}$$

$$q_w^H = \begin{cases} 1, \text{if the human disassembly workstation } w \text{ is turned on} \\ 0, \text{otherwise} \end{cases}$$

$l_{i,k,w}$: start time of task i at order k of workstation w

2) Mathematical Model:

$$max\left(\sum_{i\in\mathbb{I}}\sum_{j\in\mathbb{J}}\sum_{k\in\mathbb{K}}\sum_{w\in\mathbb{W}} p_j d_{ji} x_{i,k,w} - \sum_{i\in\mathbb{I}} c_i^H h_i - \sum_{i\in\mathbb{I}} c_i^R r_i - \sum_{w\in\mathbb{W}} q_w^R c_w^R - \sum_{w\in\mathbb{W}} q_w^H c_w^H\right) \quad (1)$$

The objective function (1) aims to maximize the disassembly profit. $\sum_{i\in\mathbb{I}}\sum_{j\in\mathbb{J}}\sum_{k\in\mathbb{K}}\sum_{w\in\mathbb{W}} p_j d_{ji} x_{i,k,w}$ represents the value gained by disassembling subassembly j. $\sum_{i\in\mathbb{I}} c_i^H h_i$ represents the cost of a human operator to perform task i. $\sum_{i\in\mathbb{I}} c_i^R r_i$ represents the cost of a robot to perform task i. $\sum_{w\in\mathbb{W}} q_w^R c_w^R$ represents the cost of turning on the robot workstation w. $\sum_{w\in\mathbb{W}} q_w^H c_w^H$ represents the cost of turning on the human workstation w.

3) Constraints:

$$\sum_{k\in\mathbb{K}}\sum_{w\in\mathbb{W}} x_{i,k,w} \le 1, i \in \mathbb{I} \tag{2}$$

$$\sum_{i\in\mathbb{I}} x_{i,k,w} \le 1, k \in \mathbb{K}, w \in \mathbb{W} \tag{3}$$

$$\sum_{i\in\mathbb{I}} x_{i,k,w} \ge \sum_{i\in\mathbb{I}} x_{i,k+1,w}, k \in \mathbb{K}\setminus K, w \in \mathbb{W}, \tag{4}$$

$$\sum_{k\in\mathbb{K}}\sum_{w\in\mathbb{W}} x_{i,k,w} + \sum_{k\in\mathbb{K}}\sum_{w\in\mathbb{W}}\sum_{i'\in\mathbb{I}_i^G} x_{i',k,w} \le 1, i \in \mathbb{I} \tag{5}$$

$$x_{i,k,w} \le \sum_{i'\in\mathbb{I}_i^F}\sum_{k'\in\mathbb{K}}\sum_{w'\in\mathbb{W}} x_{i',k',w'}, i \in \mathbb{I}, k \in \mathbb{K}, w \in \mathbb{W} \tag{6}$$

$$s_w = q_w^H + q_w^R, w \in \mathbb{W} \tag{7}$$

$$s_w \ge s_{w+1}, w \in \mathbb{W}\setminus W \tag{8}$$

$$\sum_{i\in\mathbb{I}}\sum_{k\in\mathbb{K}} x_{i,k,w} \le M s_w, w \in \mathbb{W} \tag{9}$$

$$\sum_{i\in\mathbb{I}}\sum_{k\in\mathbb{K}} x_{i,k,w} \ge s_w, w \in \mathbb{W} \tag{10}$$

$$\begin{aligned} l_{i,k,w} \ge l_{i',k',w} + t_{i'}^R r_{i'} + M(r_i + r_{i'} + x_{i,k,w} \\ + x_{i',k',w} - 4), \\ i \in \mathbb{I}^R, i' \in \mathbb{I}^R, k \in \mathbb{K}, k' \in \mathbb{K}, k' < k, w \in \mathbb{W} \end{aligned} \tag{11}$$

$$\begin{aligned} l_{i,k,w} \ge l_{i',k',w} + t_{i'}^H h_{i'} + M(h_i + h_{i'} + x_{i,k,w} \\ + x_{i',k',w} - 4), \\ i \in \mathbb{I}^H, i' \in \mathbb{I}^H, k \in \mathbb{K}, k' \in \mathbb{K}, k' < k, w \in \mathbb{W} \end{aligned} \tag{12}$$

$$\begin{aligned} l_{i,k,w} \ge l_{i',k',w} + t_{i'}^H h_{i'} + t_{i'}^R r_{i'} + M(h_i + h_{i'} \\ + r_i + r_{i'} - 2), \\ i \in \mathbb{I}, i' \in \mathbb{I}_i^F, k \in \mathbb{K}, k' \in \mathbb{K}, k' < k, w \in \mathbb{W} \end{aligned} \tag{13}$$

$$\begin{aligned} l_{i',k+1,w} \ge l_{i,k,w}, i \in \mathbb{I}, i' \in \mathbb{I}, k \in \mathbb{K}\setminus K, \\ w \in \mathbb{W} \end{aligned} \tag{14}$$

$$h_i = 0, i \in \mathbb{I}^R \tag{15}$$

$$r_i = 0, i \in \mathbb{I}^H \tag{16}$$

$$h_i + r_i = \sum_{k\in\mathbb{K}}\sum_{w\in\mathbb{W}} x_{i,k,w}, i \in \mathbb{I} \tag{17}$$

$$h_i \ge 0, i \in \mathbb{I} \tag{18}$$

$$r_i \ge 0, i \in \mathbb{I} \tag{19}$$

$$x_{i,k,w}, q_w^R, q_w^H, S_w \in \{0,1\}, i \in \mathbb{I}, k \in \mathbb{K}, w \in \mathbb{W} \tag{20}$$

$$l_{i,k,w} \ge 0, i \in \mathbb{I}, k \in \mathbb{K}, w \in \mathbb{W} \tag{21}$$

Constraint (2) ensures that each task is performed only once during disassembly. Constraint (3) limits each workstation to performing one disassembly task at a time. Constraint (4) ensures sequential task assignment, such that if a task is in a later position, a task must also be assigned to the earlier position. Constraint (5) prevents conflicting tasks from being executed simultaneously, allowing at most one to be performed. Constraint (6) enforces task precedence. Constraint (7) specifies that a workstation can be activated only as a robot or human station. Constraint (8) requires workstations to be activated sequentially. Constraint (9) restricts task assignments to already activated workstations, with M being sufficiently large to satisfy the constraint. Constraint (10) guarantees that activated workstations are assigned at least one disassembly task.

Constraint (11) ensures that when a robot performs a disassembly task, the subsequent task starts after the completion of its predecessor. This holds if both task i and task i' are performed by the robot. Constraint (12) ensures that when a human performs a disassembly task, the subsequent task starts after the completion of its predecessor. Constraint (13) defines the execution time relationship between tasks that satisfy the precedence constraint. This holds when any two of h_i, h_i', r_i, and r_i' equal one. Constraint (14) defines the sequential execution time constraint between adjacent disassembly tasks. Constraint (15) mandates that a robot perform the task. Constraint (16) mandates that a human perform the task. Constraint (17) ensures that only one person or robot can perform task i. Constraints (18) and (19) define the range of the decision variables.

III. Proposed Algorithm

A. Improved Grey Wolf Optimizer

Grey wolves are social animals known for team behaviors and strong survival abilities. A grey wolf pack follows a strict hierarchy, containing one α wolf, one β wolf, one δ wolf, and many ω wolves. Each wolf represents a solution to the optimization problem, with higher-ranked wolves typically providing better solutions than lower-ranked ones. The α wolf holds the highest rank in the pack, leading with the greatest dominance and representing the optimal solution. If a β or δ wolf achieves better fitness than the α wolf, it replaces the latter as the new leader.

The β wolf ranks second to the α wolf and is its most promising competitor. It assists the α wolf in the search process and can be promoted to α or replaced by another wolf based on fitness.

A δ wolf, weaker than a β wolf but with stronger search ability, excels at exploring new search spaces to find better solutions. If a δ wolf finds a high-quality solution, it may replace a higher-ranked wolf. It may degrade into a ω wolf if it performs poorly.

ω wolves occupy the lowest rank in the pack and are dominated by the other wolves. Due to their poor fitness, they may be eliminated. They improve their fitness mainly by observing and mimicking other wolves' behavior. Despite their subordinate status, they play a crucial role in ensuring diversity. They make up the largest proportion of the pack, maintaining population diversity.

Four grey wolves assume distinct roles in a pack, engaging in dynamic competition and cooperation. This dynamic helps the pack maintain stability and adapt to environmental changes.

This work transforms the grey wolves' prey-searching process into finding the optimal solution for the objective function (1). The algorithm operates in three stages: population initialization, hunting, and population update. The flowchart is shown in Fig. 2. The population initialization generates initial individuals and adjusts them to feasible solutions. The hunting phase calculates the fitness of the grey wolves and selects the top three solutions. The population update saves the optimal solution from each iteration to guide the subsequent search. The specific implementations are described below.

Unlike the standard GWO, the proposed IGWO incorporates a three-stage encoding scheme and a two-step search and attack mechanism. The design is inspired by balancing exploration and exploitation in complex combinatorial spaces. The theoretical basis lies in maintaining population diversity through ω wolves while guiding convergence via α, β, and δ wolves, which effectively prevents premature convergence and enhances global search capability. This combination addresses the NP-hardness of large-scale disassembly problems.

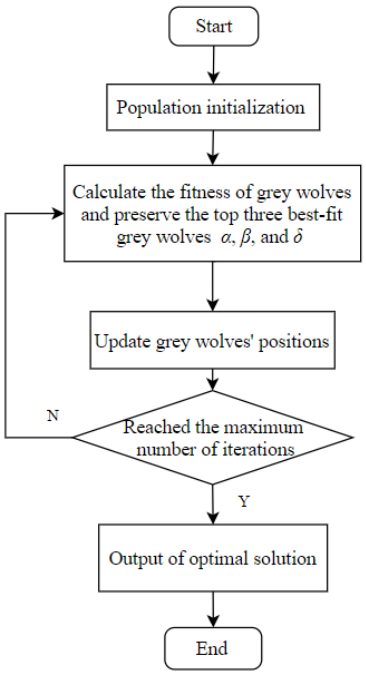


Fig. 2. The flow chart of IGWO.

B. Encoding

In this process, selecting and disassembling tasks, assigning task sequences to workstations, and allocating robots and humans influence the objective value. We design a three-stage encoding: $\theta = (\theta_1, \theta_2, \theta_3)$. θ_1 represents the sequence of disassembly tasks. θ_2 represents the workstation assigned to each disassembly task in θ_1. $\theta_3 = \{(\mu_{hr}, \mu_w)\}$ represents the assignment of humans and robots to workstations, where μ_{hr} indicates either a human or robot and μ_w denotes the workstation number.

Fig. 3 visualizes the prioritization relationship of disassembly tasks [57]. Fig. 4 combines the task, workstation, and operator (i.e., human or robot) assignment diagrams. For example, θ_1 is $\langle 1, 2, 3, 5, 9, 12, 11, 15\rangle$. The corresponding θ_2 is $\theta_2 = \langle 1, 2, 2, 1, 1, 2, 2, 1\rangle$. Tasks 1, 5, 9, and 15 are assigned to the first workstation, and tasks 2, 3, 12, and 11 to the second. Since the robot's position is fixed, there must be a workstation where the robot performs tasks. The total cost of the current disassembly sequence is calculated, revealing that tasks in the first workstation are more cost-effective when performed by the robot than by the worker. The second workstation is more suitable for worker tasks. Therefore, the robot is assigned to the first workstation and the worker to the second. $\theta_3 = \{(R, 1), (H, 2)\}$, where R represents the robot and H the human. This encoding strategy results in an actual disassembly assignment scheme for the circular disassembly line shown in the figure.

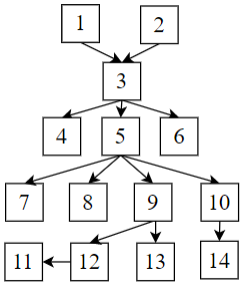


Fig. 3. Task precedence graph.

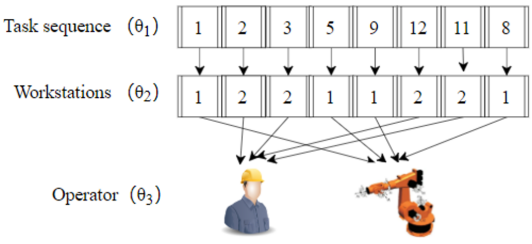


Fig. 4. Three-stage encoding example.

C. Population Initialization

The initial population is generated by random sampling. This ensures solution randomness and facilitates subsequent

searches. However, the randomly generated solutions may not be feasible. We revise these infeasible solutions to make them possible. This work categorizes tasks based on their attributes and uses a collaborative human-robot disassembly approach. Tasks for human disassembly are assigned to one workstation, while tasks for robots are assigned to another. This allocation principle enhances disassembly efficiency, reduces costs, and optimizes the process. Before allocation, tasks are defined as dangerous, delicate, or common. Dangerous tasks are assigned to robots, delicate tasks to humans, and common tasks to either. Tasks are first assigned based on their attributes, grouped by operator type, and then transferred to workstations.

Algorithm 1 Population initialization

Input: population size
Output: population P
1: **while** (n<population size) **do**
2: Generate a random task sequence θ_1
3: Adjust θ_1 to satisfy conflict and precedence relationships
4: Obtain corresponding θ_2 and θ_3 according to the encoding strategy
5: Add $\theta(\theta_1, \theta_2, \theta_3)$ to P
6: $n = n + 1$
7: **end while**
8: **return** P

D. Hunting

Grey wolves hunt through two processes: searching and attacking prey. The prey's position represents a potential solution in the improved grey wolf algorithm (IGWO) applied to the proposed problem. During hunting, the fitness of the grey wolves is evaluated. Grey wolves are ranked by fitness, and the top three (α, β, and δ) are selected as leaders. ω wolves follow the α, β, and δ wolves to explore the solution space. Grey wolves adjust their positions during searching, moving towards potential prey to find better solutions. Fig. 5 illustrates the searching process of grey wolves, showing the change in a wolf's position from π_1 to π_2. This helps the algorithm escape local optima, facilitating a global search for better solutions and enhancing population diversity.

Due to space limitations, we present a simplified search process. Initially, task 9 is randomly selected from sequence π_1. Following the precedence relations in Fig. 3, tasks 12 and 11, which follow task 9, are excluded. Next, a new subsequence, tasks 7, 10, and 14, is selected based on the precedence relation starting with task 5. This adjustment results in a new position, π_2. During the search, ω wolves follow the lead of α, β, and δ wolves. Role changes among α, β, δ, and ω wolves may occur due to differences in their search spaces. For example, if the β wolf finds prey with better fitness than the α wolf, it replaces the α wolf.

When a potential prey is encountered during the search, the grey wolf evaluates its fitness against the current optimal solution (i.e., the α wolf's fitness). If the prey's fitness surpasses the current optimal solution's fitness, the wolf prepares to attack it. Otherwise, the search continues for a better solution. Fig. 6 illustrates the attacking process. "Grey wolf" represents an individual selected from α, β, or δ, and "Prey" refers to one of the better solutions found so far. A "Mask" sequence is randomly generated with a length equal to the number of tasks. This sequence determines the source of task information: if the corresponding bit in "Mask" is 0, the information comes from the Grey wolf; otherwise, it comes from the Prey. Task assignments are continuously checked for feasibility to avoid redundancy throughout this process.

After identifying a promising solution (prey), the Grey wolf signals other wolves to join the attack, completing the hunting process. As the wolves approach the prey, those receiving the signal adjust their positions based on location. Fig. 7 shows an example of position variation. In the "Original position" sequence, task x is randomly selected. The immediately preceding task y and following task z of x are determined based on the precedence relation matrix. A "New position" (prey position) is generated by inserting task x randomly between tasks y and z. Once the prey is attacked, its fitness replaces that of the α wolf, becoming the new optimal solution. The search directions of α, β, and δ wolves are updated to guide ω wolves in exploring new spaces. This process aims to discover higher-quality solutions and converge towards the prey position, representing the optimal solution.

E. Population Update

The search process in IGWO is dynamic, with the optimal solution continuously evolving. Each iteration involves selecting the best individual from the current population based on fitness. This individual is retained in the population. In each iteration, the optimal individual is compared with others in the population. New α, β, and δ wolves are selected based on their fitness to lead the pack, updating the positions of the other wolves accordingly.

IV. Experimental Results and Evaluations

A. Experimental Setup

Table I presents the names of the test cases and the corresponding number of tasks in each case. To verify the correctness of the mathematical model, we have applied CPLEX as an exact solver that is commercially available. Next, we solve the proposed problem using IGWO. We then compare

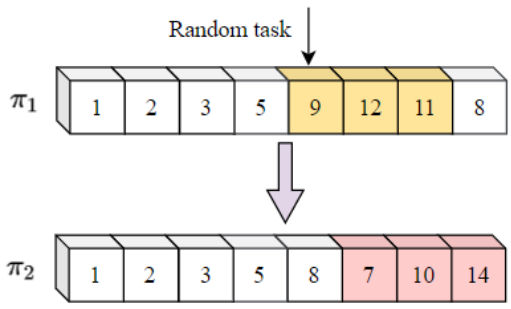


Fig. 5. Searching process.

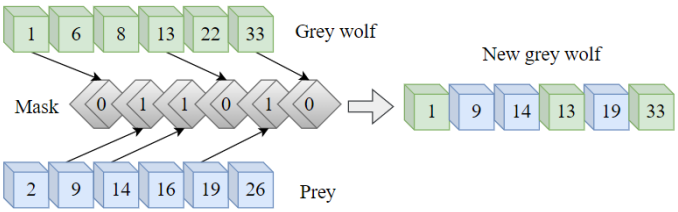


Fig. 6. Attacking process.

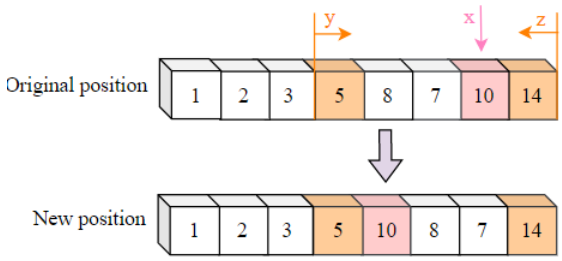


Fig. 7. Position variation during prey attack.

the obtained solution with other intelligent optimization algorithms to assess its performance. All of these operations are being performed on a Windows 10 system with an Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40 GHz environment.

We develop a mathematical planning model for human-robot collaboration circular disassembly lines. To validate and solve the model, we use the data provided in Table I to test the mathematical model developed [11, 58]. These test cases include small, medium, and large-scale instances, which have been carefully chosen to reflect different levels of problem complexity. The consistent superiority over other algorithms across these cases highlights its scalability and effectiveness in solving complex disassembly optimization problems.

B. Experimental Results

Table II summarizes the results of validating our mathematical model using CPLEX. The table presents case ID, task assignments, maximum disassembly profit, and computational time. In the "Assignment of tasks" column, symbols *H* and *R* represent the human and robot disassembly workstations, respectively. For example, in Case 1, Tasks 1, 3, and 7 are assigned to the first workstation, where a robot performs the disassembly tasks. Task 10 is allocated to the second workstation, where a human handles the disassembly. Cases 1 to 3, detailed in Table I, represent small and medium-sized scenarios where CPLEX identifies optimal sequences and solutions. However, for larger cases (Cases 4 to 6), CPLEX fails to find an optimal solution within the 7200-second time limit. This limitation highlights the computational challenges of NP-hard problems in large-scale disassembly scenarios. In summary, the results in Table II demonstrate the feasibility and accuracy of our mathematical model.

Table III presents the results of applying IGWO to each case. This table shows that IGWO finds optimal solutions quickly for small and large cases. As expected, the solution time increases with case size; however, the algorithm consistently delivers high-quality solutions across all scenarios. This demonstrates the robustness of IGWO in handling disassembly optimization problems, balancing computational efficiency with solution quality, even as complexity increases.

Table IV compares the performance of IGWO and CPLEX across different case sizes. In solving single-objective problems, CPLEX achieves the known optimal solution. For small and medium-sized cases (Cases 1-3), IGWO consistently matches the global optimal solutions obtained by CPLEX, but in significantly less time. Specifically, IGWO is 2 to 30 times faster than CPLEX for these cases. In contrast, for large cases (Cases 4-6), CPLEX fails to find a solution within the 7200-second time limit, while IGWO finds feasible solutions in approximately 25 seconds. This significant performance difference highlights IGWO's ability to solve larger-scale disassembly optimization problems efficiently. The "Improvement" column in the table shows the speed advantage of IGWO over CPLEX. IGWO's advantage becomes more pronounced as case size increases, highlighting its effectiveness and efficiency in handling complex disassembly optimization tasks. These results confirm that IGWO is a highly effective alternative to CPLEX, particularly suited for large-scale disassembly optimization where timely solutions are crucial.

Fig. 8 and Fig. 9 illustrate the task allocation for circular and linear disassembly line layouts with equal task sizes. The disassembly process includes hazardous tasks, which can incur high costs for the employer and the employee if injured during execution [59]. Hazardous tasks in Case 4 include <1, 6, 21, 40>. The figures show that linear and circular disassembly lines assign robots to hazardous tasks to ensure safer disassembly. The key difference between the two layouts lies in the number of humans and robots required to complete the same task size. The linear layout requires four humans and four robots to work simultaneously, while the circular layout requires only two humans and two robots. This layout

TABLE I Case information

Case ID	Case name	Number of tasks
1	Washing machine	13
2	Treadmill	17
3	Recirculating ball steering machine	21
4	Microwave Oven	44
5	Knotting machine	52
6	Refrigerator	66

TABLE II CPLEX solutions

Case ID	Assignment of tasks	Maximum profit	Running time
1	(1,3,7)→r,(10)→h	1090	3.008s
2	(2,14,6,12,16)→r,(17)→h	1290	130.024s
3	(1,8,5)→r,(3,13,20,15,17,19)→h	2189	53.051s
4	———-	——	7200s
5	———-	——	7200s
6	———-	——	7200s

TABLE III IGWO solutions

Case ID	Assignment of tasks	Maximum profit	Running time
1	(1,3,7)→r,(10)→h	1090	1.43s
2	(2,6,14,12,16)→r,(17)→h	1290	4.389s
3	(1,8,5)→h,(3,13,15,17,20,19)→r	2189	4.732s
4	(1,3,19,10,28,29,30,20,13,31,39,40,41,26,8,9,21,11,6)→r, (7,18,44,42,2,35,23,4,22,25,27,12,32,33,5,43,37,38,14,15)→h	1045	12.688s
5	(36,31,18,49,51,26,1,9,16,21,29,11,46,30,40,41,38,43,39,8,13,10,6,20,50,48)→r, (4,25,3,32,45,52,12,14,33,47,23,28,15,44,37,27,34,22,35,7,24,42,17,5)→h	2983	8.646s
6	(1,9,50,10,56,61,23,26,29,53,36,6,39,51,19,30,20,49,40,41, 21,46,59,16,31,11,66,60,38)→r, (35,22,44,2,45,54,3,55,57,24,64,25,62,52,12,65,27,13,28,14,7,5,32,33,48,63,34,58,15,42,8,17,47,43,37,4,18)→h	2291	24.82s

TABLE IV Comparison between IGWO and CPLEX

Case ID	Maximum profit		Running time		
	CPLEX	IGWO	CPLEX	IGWO	Improvement
1	**1090**	1090	3.008s	1.430s	2.103s
2	1290	1290	130.024s	4.389s	29.625s
3	2189	2189	53.051s	4.732s	11.211s
4	—	1045	—	12.688s	—
5	—	2983	—	8.646s	—
6	—	2291	—	24.82s	—

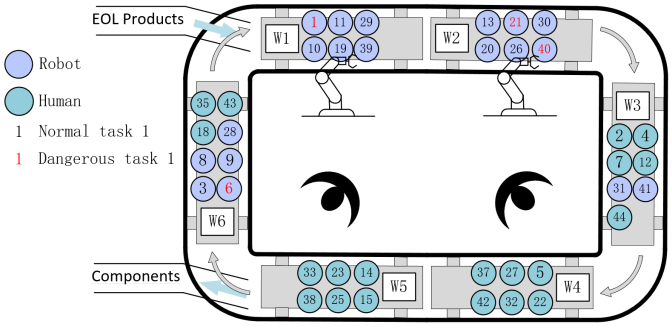


Fig. 8. Circular disassembly line disassembly Case 4.

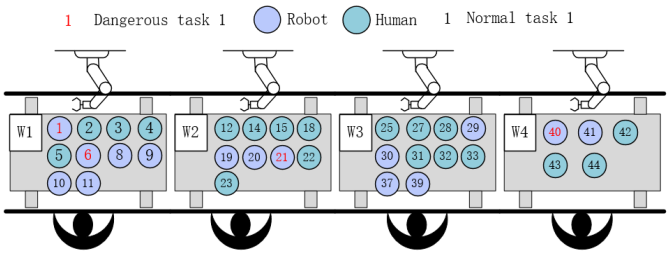


Fig. 9. Linear disassembly line disassembly Case 4.

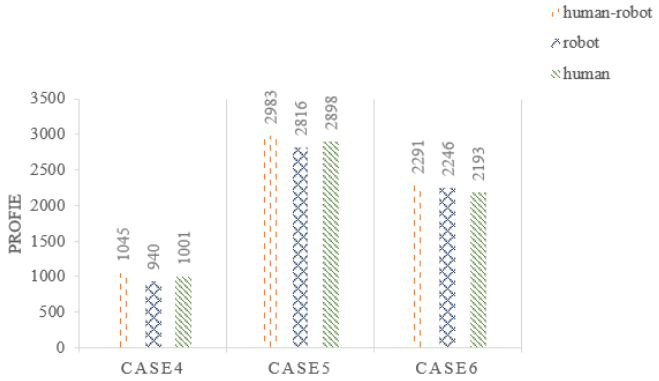


Fig. 10. Comparison of three disassembly models.

optimizes spatial utilization and reduces operational costs.

In contrast, task assignments in a linear disassembly layout are primarily guided by precedence relations, with additional constraints from cycle time and space limitations. This structure requires activating new workstations when tasks exceed cycle times, dispersing tasks across multiple stations to avoid overloading any single operator. While this methodical approach ensures adherence to operational constraints, it may reduce efficiency in distributing tasks that benefit from operator specialization. The circular disassembly layout overcomes these constraints by using a cyclic assignment mechanism. This mechanism consolidates tasks suited to the same operator on a single workstation, improving operational efficiency and profitability. Unlike the linear layout, the circular layout optimally balances task distribution and operator specialization, providing a superior solution for complex disassembly operations.

Fig. 10 compares three distinct disassembly models: human-robot collaborative, human-only, and robot-only. The graph plots cases on the horizontal axis and maximum profit on the vertical axis, with each disassembly mode represented by a different colored line. The orange line represents human-robot collaborative disassembly, the blue line represents robot-only disassembly, and the green line represents human-only disassembly. Examination of the graph reveals that human-robot collaborative disassembly offers notable advantages over robot-only and human-only modes. Specifically, the collaborative approach achieves higher disassembly efficiency and maximizes profit across all analyzed cases. This advantage highlights the synergistic benefits of combining human precision with robotic efficiency in disassembly operations.

Table V presents a comparative analysis of IGWO and other algorithms—CPA, SSA, WOA, and FOA—applied to our disassembly cases. These algorithms use distinct search strategies and behavioral patterns, providing a comprehensive performance overview across different cases. IGWO consis-

TABLE V Algorithm Comparison

Case	IGWO		CPA		SSA		WOA		FOA	
	Profit	Running time	Profit	Running time	Profit	Running time	Profit	Running time	Profit	Running time
1	**1090**	1.430s	**1090**	2.229s	**1090**	23.390s	1027	4.04s	**1090**	6.158s
3	**1290**	4.389s	**1290**	2.863s	**1290**	26.341s	1284	4.302s	**1290**	5.849s
3	**2189**	4.732s	**2189**	3.029s	**2189**	26.553s	2186	4.912s	**2189**	11.525s
4	**1045**	12.688s	1019	11.453s	1037	26.189s	1019	7.460s	1019	47.102s
5	**2983**	8.646s	2973	8.076s	2973	22.004s	2973	5.667s	2973	75.965s
6	**2291**	24.82s	**2291**	24.825s	**2291**	22.970s	**2291**	11.983s	**2291**	92.258s

tently outperforms the other algorithms in maximizing profit, particularly in Cases 4 and 5. While solving time increases with problem size for all algorithms, IGWO generally outperforms SSA and FOA, competes closely with CPA, and shows negligible differences compared to WOA. In conclusion, IGWO is a highly effective algorithm for solving CCDP, demonstrating robustness across various problem sizes and outperforming alternative optimization approaches regarding solution quality.

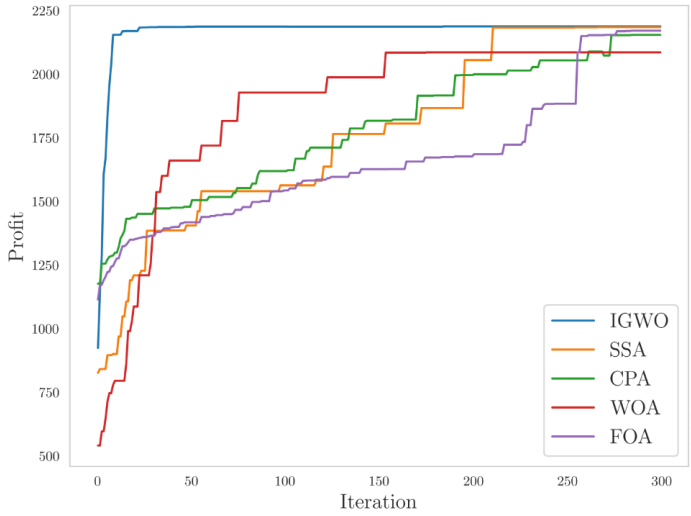


Fig. 11. Comparison of algorithm convergence curves for Case 3.

We conduct 20 independent experiments for each algorithm. The average experimental results are calculated, and the convergence curve is plotted. As shown in Fig. 11, IGWO finds the optimal solution more quickly and demonstrates superior convergence performance compared to the four other algorithms. Therefore, IGWO emerges as the superior algorithm for CCDP in this study. Specifically, it exhibits superior convergence performance and accuracy compared to its peers in solving large-scale combinatorial optimization problems.

This work introduces IGWO as a novel approach to solving CCDP. A mathematical model is developed to maximize profit in CCDP. We use CPLEX to validate the correctness of the mathematical model and verify IGWO's solution capability. Experimental results confirm IGWO's superiority in solving CCDP. Specifically, we simulate the searching and attacking processes of grey wolves. The searching process enables grey wolves to expand their search range, avoid local optima, and discover higher-quality solution spaces.

V. Conclusion and Future Work

This work introduces IGWO as a novel approach to solving CCDP. A mathematical model is developed to maximize profit in CCDP. We use CPLEX to validate the competency of the mathematical model and verify IGWO's solution capability. The superiority of IGWO in solving CCDP is confirmed through experimental results. Specifically, we simulate the searching and attacking processes of grey wolves. The searching process enables grey wolves to expand their search range, avoid local optima, and discover higher-quality solution spaces.

Our next work aims to 1) apply IGWO to solving multi-objective CCDP, 2) improve the flexibility of human-robot collaborative disassembly, and 3) improve the algorithm and apply it to handle more complex problems in real-world manufacturing and remanufacturing contexts.

In practical deployment, the proposed IGWO can be integrated into manufacturing execution systems or digital twin platforms, enabling adaptive scheduling in real-time. However, further collaboration with industry partners and on-site testing will be essential to validate scalability and robustness under real production conditions.

References

[1] T. Brogårdh, "Present and future robot control development—an industrial perspective," *Annual Reviews in Control*, vol. 31, no. 1, pp. 69–79, 2007.

[2] R. Li, W. Wang, Y. Chen, S. Srinivasan, and V. N. Krovi, "An end-to-end fully automatic bay parking approach for autonomous vehicles," in *Dynamic Systems and Control Conference*, vol. 51906. American Society of Mechanical Engineers, 2018, p. V002T15A004.

[3] W. Wang, Z. Przedworska, J. Parron, M. Lyons, M. Zhu, and A. Tuininga, "Mcros: A multimodal collaborative robot system for human-centered tasks," in *2024 International Conference on Networking, Sensing and Control (ICNSC)*. IEEE, 2024, pp. 1–6.

[4] S. Bier, R. Li, and W. Wang, "A full-dimensional robot teleoperation platform," in *2020 11th International Conference on Mechanical and Aerospace Engineering (ICMAE)*. IEEE, 2020, pp. 186–191.

[5] 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*, vol. 10, no. 4, pp. 2061–2073, 2022.

[6] X. Guo, M. Zhou, A. Abusorrah, F. Alsokhiry, and K. Sedraoui, "Disassembly sequence planning: a survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 7, pp. 1308–1324, 2020.

[7] A. Priyono, W. Ijomah, and U. S. Bititci, "Disassembly for remanufacturing: A systematic literature review, new model development and future research needs," *Journal of Industrial Engineering and Management (JIEM)*, vol. 9, no. 4, pp. 899–932, 2016.

[8] X. Guo, Z. Bi, J. Wang, S. Qin, S. Liu, and L. Qi, "Reinforcement learning for disassembly system optimization problems: A survey," *International Journal of Network Dynamics and Intelligence*, vol. 2, no. 1, pp. 1–14, 2023.

[9] E. Zussman and M. Zhou, "A methodology for modeling and adaptive planning of disassembly processes," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 1, pp. 190–194, 1999.

[10] G. Xu, Z. Zhang, Z. Li, X. Guo, L. Qi, and X. Liu, "Multi-objective discrete brainstorming optimizer to solve the stochastic multiple-product robotic disassembly line balancing problem subject to disassembly failures," *Mathematics*, vol. 11, no. 6, p. 1557, 2023.

[11] E. Özceylan, C. B. Kalayci, A. Güngör, and S. M. Gupta, "Disassembly line balancing problem: a review of the state of the art and future directions," *International Journal of Production Research*, vol. 57, no. 15-16, pp. 4805–4827, 2019.

[12] C. B. Kalayci, O. Polat, and S. M. Gupta, "A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem," *Annals of Operations Research*, vol. 242, pp. 321–354, 2016.

[13] X. Guo, M. Zhou, S. Liu, and L. Qi, "Lexicographic multiobjective scatter search for the optimization of sequence-dependent selective disassembly subject to multiresource constraints," *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 3307–3317, 2019.

[14] X. Cui, X. Guo, M. Zhou, J. Wang, S. Qin, and L. Qi, "Discrete whale optimization algorithm for disassembly line balancing with carbon emission constraint," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 3055–3061, 2023.

[15] Z. Li, I. Kucukkoc, and Z. Zhang, "Iterated local search method and mathematical model for sequence-dependent u-shaped disassembly line balancing problem," *Computers and Engineering, Industrial*, vol. 137, p. 106056, 2019.

[16] Z. A. Cil, D. Kizilay, Z. Li, and H. J. E. S. w. A. Oeztop, "Two-sided disassembly line balancing problem with sequence-dependent setup time: A constraint programming model and artificial bee colony algorithm," *Expert Systems with Applications*, vol. 203, p. 117529, 2022.

[17] K. Wang, X. Li, L. Gao, P. Li, and S. M. Gupta, "A genetic simulated annealing algorithm for parallel partial disassembly line balancing problem," *Applied Soft Computing*, vol. 107, p. 107404, 2021.

[18] Y. Frein, C. Commault, and Y. Dallery, "Analytical performance evaluation of closed transfer lines with limited number of pallets," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*. IEEE Computer Society, 1992, pp. 1018–1019.

[19] S. Biller, S. P. Marin, S. M. Meerkov, and L. Zhang, "Closed bernoulli production lines: Analysis, continuous improvement, and leanness," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 1, pp. 168–180, 2009.

[20] A. Angius, M. Colledani, A. Horváth, and S. B. Gershwin, "Analysis of the lead time distribution in closed loop manufacturing systems," *International Federation Of Automatic Control PapersOnLine*, vol. 49, no. 12, pp. 307–312, 2016.

[21] A. Alduaij and N. M. Hassan, "Adopting a circular open-field layout in designing flexible manufacturing systems," *International Journal of Computer Integrated Manufacturing*, vol. 33, no. 6, pp. 572–589, 2020.

[22] P. Hamet and J. Tremblay, "Artificial intelligence in medicine," *Metabolism*, vol. 69, pp. S36–S40, 2017.

[23] W. Wang, R. Li, Y. Chen, Z. M. Diekel, and Y. Jia, "Facilitating human–robot collaborative tasks by teaching-learning-collaboration from human demonstrations," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 640–653, 2018.

[24] W. Wang, R. Li, Z. M. Diekel, Y. Chen, Z. Zhang, and Y. Jia, "Controlling object hand-over in human–robot collaboration via natural wearable sensing," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 1, pp. 59–71, 2019.

[25] G. Modery and W. Wang, "Assisting humans in human-robot collaborative assembly contexts through deep q-learning," in *2024 IEEE MIT Undergraduate Research Technology Conference (URTC)*. IEEE, 2024, pp. 1–6.

[26] S. Qin, S. Zhang, J. Wang, S. Liu, X. Guo, and L. Qi, "Multiobjective multiverse optimizer for multirobotic u-shaped disassembly line balancing problems," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 2, pp. 882–894, 2023.

[27] Z. A. Çil, S. Mete, and F. Serin, "Robotic disassembly line balancing problem: A mathematical model and ant colony optimization approach," *Applied Mathematical Modelling*, vol. 86, pp. 335–348, 2020.

[28] H. Diamantopoulos and W. Wang, "Accommodating and assisting human partners in human-robot collaborative tasks through emotion understanding," in *2021 12th International Conference on Mechanical and Aerospace Engineering (ICMAE)*. IEEE, 2021, pp. 523–528.

[29] C. Hannum, R. Li, and W. Wang, "A trust-assist framework for human–robot co-carry tasks," *Robotics*, vol. 12, no. 2, p. 30, 2023.

[30] P. Tilloo, J. Parron, O. Obidat, M. Zhu, and W. Wang, "A pomdp-based robot-human trust model for human-robot collaboration," in *2022 12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2022, pp. 1009–1014.

[31] M.-L. Lee, S. Behdad, X. Liang, and M. Zheng, "Task allocation and planning for product disassembly with human–robot collaboration," *Robotics and Computer-Integrated Manufacturing*, vol. 76, p. 102306, 2022.

[32] K. Li, Q. Liu, W. Xu, J. Liu, Z. Zhou, and H. Feng, "Sequence planning considering human fatigue for human-robot collaboration in disassembly," *Procedia Conference on Industrial Product-Service Systems*, vol. 83, pp. 95–104, 2019.

[33] T. Wu, Z. Zhang, Y. Zeng, and Y. Zhang, "Mixed-integer programming model and hybrid local search genetic algorithm for human-robot collaborative disassembly line balancing problem," *International Journal of Production Research*, vol. 62, no. 5, pp. 1758–1782, 2024.

[34] P. Tsarouchi, A. S. Matthaiakis, S. Makris, and G. Chryssolouris, "On a human-robot collaboration in an assembly cell," *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 6, pp. 1–10, 2016.

[35] Z. Liu, Q. Liu, W. Xu, Z. Liu, Z. Zhou, and J. Chen, "Deep learning-based human motion prediction considering context awareness for human-robot collaboration in manufacturing," *Procedia Conference on Industrial Product-Service Systems*, vol. 83, pp. 272–278, 2019.

[36] I. Belhadj, M. Aicha, and N. Aifaoui, "Product disassembly planning and task allocation based on human and robot collaboration," *International Journal on Interactive Design and Manufacturing*, vol. 16, no. 2, pp. 803–819, 2022.

[37] X. Guo, S. Liu, M. Zhou, and G. Tian, "Dual-objective program and scatter search for the optimization of disassembly sequences subject to multiresource constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1091–1103, 2017.

[38] L. Ding, Y. Feng, J. Tan, and Y. Gao, "A new multi-objective ant colony algorithm for solving the disassembly line balancing problem," *The International Journal of Advanced Manufacturing Technology*, vol. 48, pp. 761–771, 2010.

[39] X. Guo, C. Fan, M. Zhou, S. Liu, J. Wang, S. Qin, and Y. Tang, "Human–robot collaborative disassembly line balancing problem with stochastic operation time and a solution via multi-objective shuffled frog leaping algorithm," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 3, pp. 4448 – 4459, 2024.

[40] Y. Ren, D. Yu, C. Zhang, G. Tian, L. Meng, and X. Zhou, "An improved gravitational search algorithm for profit-oriented partial disassembly line balancing problem," *International Journal of Production Research*, vol. 55, no. 23-24, pp. 7302–7316, 2017.

[41] C. B. Kalayci and S. M. Gupta, "Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7231–7241, 2013.

[42] X. Wang, F. Chu, T. Ren, and D. Bai, "Bi-objective optimization of a flow shop scheduling problem under time-of-use tariffs," *IEEE Transactions on Automation Science and Engineering*, 2024.

[43] H. Yuan and M. Zhou, "Profit-maximized collaborative computation offloading and resource allocation in distributed cloud and edge computing systems," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1277–1287, 2021.

[44] X. Guo, L. Zhou, M. Zhou, W. Wang, J. Wang, and L. Qi, "Balancing human-robot collaborative disassembly line by using dingo optimization algorithm," in *2024 IEEE 4th International Conference on Human-Machine Systems (ICHMS)*, 2024, pp. 1–6.

[45] B. Zhou, Y. Wang, B. Zi, and W. Zhu, "Fuzzy adaptive whale optimization control algorithm for trajectory tracking of a cable-driven parallel robot," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 4, pp. 5149–5160, 2024.

[46] D. Liu, L. Shang, X. Dai, and Z. Zhang, "Modeling and solving the allocation problem of spatiotemporal crowdsourced logistics tasks in social manufacturing," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 6, pp. 7967–7975, 2024.

[47] X. Cui, X. Guo, M. Zhou, J. Wang, S. Qin, and L. Qi, "Discrete whale optimization algorithm for disassembly line balancing with carbon emission constraint," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 3055–3061, 2023.

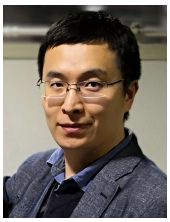
[48] J. Zhao, S. Liu, M. Zhou, X. Guo, and L. Qi, "Modified cuckoo search

algorithm to solve economic power dispatch optimization problems," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 4, pp. 794–806, 2018.
[49] Z. Zhao, M. Zhou, and S. Liu, "Iterated greedy algorithms for flow-shop scheduling problems: A tutorial," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1941–1959, 2021.
[50] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
[51] 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.
[52] P. Anitha and B. Kaarthick, "Oppositional based laplacian grey wolf optimization algorithm with svm for data mining in intrusion detection system," *Journal of Ambient Intelligence and Computing, Humanized*, vol. 12, pp. 3589–3600, 2021.
[53] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: a review of recent variants and applications," *Neural Computing and Applications*, vol. 30, pp. 413–435, 2018.
[54] Z. Bi, X. Guo, J. Wang, S. Qin, and G. Liu, "Deep reinforcement learning for truck-drone delivery problem," *MDPI Drone*, vol. 7, no. 7, p. 445, 2023.
[55] E. Emary, H. M. Zawbaa, and C. Grosan, "Experienced gray wolf optimization through reinforcement learning and neural networks," *IEEE Transactions on Neural Networks and Systems, Learning*, vol. 29, no. 3, pp. 681–694, 2017.
[56] M. H. Sulaiman, W. L. Ing, Z. Mustaffa, and M. R. Mohamed, "Grey wolf optimizer for solving economic dispatch problem with valve-loading effects," *ARPN Journal of Engineering and Applied Sciences*, vol. 10, no. 21, pp. 1619–1628, 2015.
[57] T. Yin, Z. Zhang, Y. Zhang, T. Wu, and W. Liang, "Mixed-integer programming model and hybrid driving algorithm for multi-product partial disassembly line balancing problem with multi-robot workstations," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102251, 2022.
[58] P. Nowakowski, "A novel, cost efficient identification method for disassembly planning of waste electrical and electronic equipment," *Journal of Cleaner Production*, vol. 172, 2018.
[59] Z. Xu and Y. Han, "Two sided disassembly line balancing problem with rest time of works: A constraint programming model and an improved NSGA II algorithm," *Expert Systems with Applications*, vol. 239, p. 122323, 2024.

Zhiwei Zhang received his M.S. degree in Computer and Communication Engineering at Liaoning Petrochemical University, Fushun, China in 2019. He is currently a teaching assistant of the College of Artificial Intelligence and Software at Liaoning Petrochemical University. His current research interests include remanufacturing and intelligent optimization algorithm.

Shaokang Dai received his degree in software engineering from Southeast University Chengxian College, China, in 2022. He is currently a graduate student at the School of Artificial Intelligence and Software, Liaoning Petrochemical University. His current research interests include remanufacturing, intelligent optimization algorithm, and graph neural networks.

Chong Li received her B.S. degree in Software Engineering from Liaocheng University Dongchang College, Shandong, China, in 2021. She received her MA.ENG. degree in Computer Science and Technology at Liaoning Petrochemical University, Fushun, China, in 2024. She currently has 2 technical papers published in SMC conferences and magazine proceedings. Her current research interests include multi-objective optimization, recycling and reuse, and intelligent optimization algorithms.



Weitian Wang received his Ph.D. degree from the University of Science and Technology of China in 2016. He was a postdoctoral fellow at Clemson University. He is an Associate Professor in the School of Computing and the Founder Director of Collaborative Robotics and Smart Systems Laboratory (CRoSS Lab) at Montclair State University. His research interests include collaborative robotics, artificial intelligence, smart systems, and their synergistic cross-disciplinary applications in smart manufacturing and remanufacturing, healthcare, smart ecosystem, intelligent transportation, smart agriculture, daily assistance, data science, cybersecurity, and interactive learning. He is the recipient of IEEE R1 Technological Innovation Award.

Jesse Parron is currently a research associate in the Collaborative Robotics and Smart Systems Laboratory and an instructor in the School of Computing, Montclair State University. His research interests include collaborative robotics, machine learning, artificial intelligence, and sensor technologies.

Emilio Herrera is currently a research assistant in the Collaborative Robotics and Smart Systems Laboratory at Montclair State University. His research interests include robotics, artificial intelligence, sensor technologies, and smart manufacturing and remanufacturing.