# Two-Stage Robust Optimization for Circular Disassembly Line Balancing with Budgeted Time Uncertainty

Kun Wang and ZiHao Wei

***Abstract*—The proliferation of end-of-life(EOL) electronic products necessitates efficient disassembly recycling, yet fluctuations in task times caused by equipment wear and operator variability compromise deterministic line-balancing schedules. This paper addresses the circular disassembly line balancing problem under budgeted task-time uncertainty, where workstations are arranged radially to improve space utilization and component feeding flexibility. A profit-maximizing two-stage robust optimization model is established, in which a cardinality-constrained uncertainty set controls the conservatism level against simultaneous worst-case task durations. The resulting max-max-min formulation is solved by a tailored L-shaped decomposition that separates first-stage task assignment and workstation activation from second-stage worst-case delay evaluation via strong duality. Extensive experiments on small, medium, and large-scale instances demonstrate rapid algorithmic convergence (converging within 4 iterations with a maximum computation time of 2.15 seconds) and quantify the profit-robustness trade-off as the uncertainty budget increases, revealing profit reduction rates of 11.0%, 43.1%, and 13.7% across the respective instances.**

***Index Terms*—Two-stage robust optimization, circular disassembly line balancing, budgeted uncertainty, disassembly time uncertainty, L-shaped decomposition**

## I. Introduction

The rapid proliferation of consumer electronics has precipitated a global surge in end-of-life (EOL) products, posing severe challenges to environmental sustainability. According to the 2024 Global E-waste Monitor, worldwide electronic waste generation escalated from 62 billion kilograms in 2010 to 96 billion kilograms in 2022, with projections indicating a further increase to 120 billion kilograms by 2030 [1, 2]. In manufacturing-intensive economies, centralized recycling infrastructure remains insufficient, causing hazardous e-waste to enter landfills without scientific processing. The resulting heavy metal, plastic, and chemical contamination threatens soil, water, and air quality, endangering both ecological systems and public health [3, 4, 5]. Against this backdrop, efficient disassembly recycling has emerged as a critical strategy for recovering valuable components and mitigating resource depletion. However, maximizing disassembly profit requires carefully balancing the number of activated workstations, task sequencing, and processing times—a challenge known as the disassembly line balancing problem (DLBP) [6, 7].

Existing research on DLBP has explored multiple line layouts to improve operational efficiency [8]. Early work by Ilgin et al. [9] focused on linear layouts because of their simple structure and ease of implementation. For complex products, Hezer and Kara [10] proposed parallel workstations to reduce bottleneck congestion, while Agrawal and Tiwari [11] investigated U-shaped configurations to enhance operator flexibility and material flow.Recently, circular and closed-loop layouts have gained traction in broader automated manufacturing and assembly systems due to their superior spatial efficiency and continuous material routing capabilities [12, 13]. However, successfully translating these geometric advantages to EOL disassembly lines remains a formidable challenge. Despite these advances, existing disassembly layouts predominantly assume deterministic disassembly times and rectilinear material flow, which limits spatial utilization and fails to accommodate the stochastic nature of manual disassembly [14]. More importantly, while the circular arrangement of workstations represents an intuitively appealing topology for compact factory floors, it has received limited attention in the recent DLBP literature [15], and its complex interaction with task-time uncertainty remains underexplored. This gap is significant because traditional deterministic models often become infeasible or severely suboptimal when confronted with the time fluctuations inherent in real-world disassembly environments. Despite these advances, existing layouts predominantly assume deterministic disassembly times and rectilinear material flow, which limits spatial utilization and fails to accommodate the stochastic nature of manual disassembly. More importantly, the circular arrangement of workstations—an intuitively appealing topology for compact factory floors—has received limited attention in the DLBP literature, and its interaction with task-time uncertainty remains underexplored [16]. This gap is significant because traditional deterministic models often become infeasible or severely suboptimal when confronted with the time fluctuations inherent in real-world disassembly environments.

To overcome the space and flow limitations of existing

Manuscript received May 23, 2026; revised June 20 and June 27, 2026; accepted June 28, 2026. This article was recommended for publication by Associate Editor Shujin Qin upon evaluation of the reviewers' comments.



This work is supported in part by NSFC under Grant Nos. 61573089, 62073069 and 51405075. Liaoning Province Education Department Scientific Research Foundation of China under Grant No. L2019027; Liaoning Province Dr. Research Foundation of China under Grant No. 20170520135. Liaoning Revitalization Talents Program under Grant No. XLYC1907166; The Natural Science Foundation of Shandong Province under Grant ZR2019BF004.

K. Wang and Z. Wei are with the College of Artificial Intelligence and Software, Liaoning Petrochemical University, Fushun 113001, China (e-mail: wk180558@gmail.com; 952048182@qq.com).

*Corresponding Author: Kun Wang.*

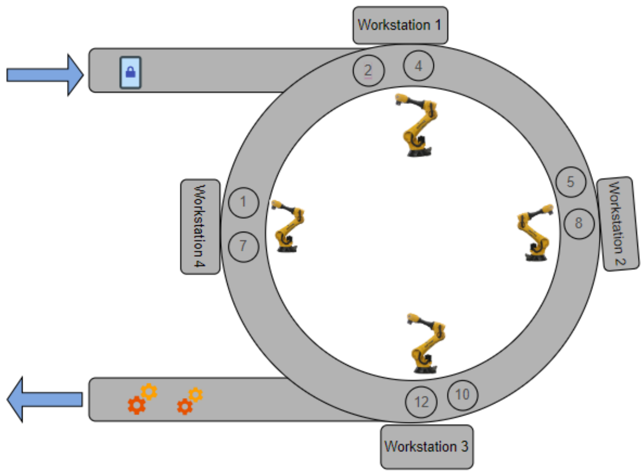



Fig. 1. The circular disassembly line layout.

topologies, this paper proposes a circular disassembly line in which workstations are arranged radially around a central hub, as illustrated in Fig. 1. Unlike rectilinear layouts, the circular topology allows components to be cyclically distributed to each station, thereby improving spatial utilization and increasing task-allocation flexibility. This geometric arrangement is particularly advantageous for compact workshops where floor space is constrained and multi-directional component feeding is desired. Nevertheless, merely changing the physical layout is insufficient: in practice, disassembly times fluctuate significantly due to equipment wear, operator skill variance, and product preservation status. Existing studies have addressed such uncertainty through stochastic programming [17, 18], fuzzy programming [19], or interval programming [20], all of which require exact probability distributions, membership functions, or precise interval bounds that are difficult to estimate from limited historical records. When the true distribution is unknown, these deterministic-equivalent approaches may yield schedules that are fragile or economically inefficient. Consequently, there is an urgent need for a distribution-free optimization framework that can simultaneously exploit the spatial advantages of circular layouts and immunize schedules against disassembly time variability.

The necessity of incorporating time uncertainty into circular disassembly line balancing stems from the interplay between operational reality and topological vulnerability. In manual disassembly, task times are inherently variable: the same operation may take several times longer on a worn device than on a well-preserved one, depending on usage history, maintenance status, and environmental conditions [21]. When these fluctuations are ignored, deterministic schedules inevitably violate cycle time constraints on the shop floor, causing workstation overload, downstream blocking, and profit erosion. The situation is exacerbated in a circular layout because workstations are radially interconnected rather than serially arranged. An overload at any station disrupts the cyclic feeding rhythm, propagating delays around the loop and amplifying waiting times for subsequent tasks [22, 23]. Unlike linear lines where bottlenecks can be isolated at a single segment, circular topologies require system-wide protection against concurrent time elongations. Moreover, deterministic models—despite their mathematical tractability—rest on nominal durations that rarely materialize in practice. By explicitly bounding deviations through a robust framework, we obtain schedules that remain feasible and profitable under worst-case realizations, thereby translating the spatial advantages of circular layouts into reliable operational performance [24].

To avoid unrealistic distributional assumptions that stochastic and fuzzy programming require, while preserving scheduling flexibility, this paper adopts two-stage robust optimization (TSRO) coupled with a budget uncertainty set. Unlike single-stage robust optimization, which commits all decisions before uncertainty realizes and consequently lacks operational adaptability, TSRO partitions decision-making into a first-stage structural commitment and a second-stage reactive adjustment [25, 26]. In the disassembly context, workstation activation and task-to-workstation assignment must be predetermined during the planning phase, whereas operational adjustments—such as absorbing delay penalties—can be made dynamically once true disassembly times are revealed. Furthermore, among uncertainty set families, budget uncertainty sets offer unique practical advantages [27]: by introducing a cardinality budget $\Gamma$, the decision maker can explicitly control the maximum number of tasks that simultaneously attain worst-case durations [28]. This not only reflects the limited simultaneity of extreme events on the shop floor, but also provides a tunable dial between robustness and optimality, allowing managers to align protection levels with their risk appetite without enumerating all scenarios [29].

The contributions of this paper are summarized as follows.

- To the best of our knowledge, we propose a profit-maximizing disassembly line balancing model for circular layouts under budgeted task-time uncertainty. The circular topology improves spatial utilization and enables cyclic component distribution, while the cardinality-constrained uncertainty set captures disassembly time variability without requiring probability distributions or fuzzy membership functions.
- Because the resulting two-stage robust model exhibits a max-max-min structure that standard MILP solvers cannot handle directly, we tailor an L-shaped decomposition procedure. The master problem determines first-stage assignment and workstation activation decisions, while the subproblem identifies worst-case delays and minimum penalties via strong duality. Iterative Benders cuts are generated to converge to the optimal solution without full scenario enumeration.
- We conduct systematic computational experiments on small(12 tasks), medium(33 tasks), and large-scale(51 tasks) benchmark instances derived from real-world product data. The results demonstrate that the tailored decomposition converges rapidly within four iterations, and quantify the explicit profit-robustness trade-off as the uncertainty budget $\Gamma$ increases.

The remainder of this paper is organized as follows. Section 2 formally describes the circular disassembly line balancing problem under budgeted time uncertainty and presents the two-stage robust optimization model. Section 3details the tailored L-shaped decomposition and subproblem dualization. Section 4 reports computational results and sensitivity analyses. Section 5 concludes the paper and suggests future research directions.

## II. Problem Description

### A. Circular Layout Description

In this study, the disassembly workstations are arranged radially around a central area, and materials flow continuously along the circular direction. Due to the closed-loop nature of this topology, Workstation 1 and Workstation $W$ are also adjacent to each other. Compared with conventional linear disassembly lines, the circular layout features cyclic material supply and coordination between adjacent workstations under limited space. The proposed model characterizes this layout mainly from the perspectives of task assignment and workstation activation. Furthermore, to mathematically express this structure without introducing additional complex spatial constraints, we define the circular distance between any two workstations $w$ and $w'$ in plain text as $D_{w,w'} = \min(|w - w'|, W - |w - w'|)$. This distance formula is used to interpret the adjacency relationship between circular workstations.

### B. Problem Statement

We consider the aforementioned circular disassembly line with $W$ workstations arranged as described above. A set $I = \{1, 2, \ldots, I\}$ of disassembly tasks must be assigned to these workstations subject to precedence constraints. Building upon this circular topology, components can circulate bidirectionally around the loop; each workstation can receive subassemblies from its two neighboring stations and forward processed parts to the next station in the cycle [30]. This cyclic feeding mechanism improves spatial utilization but introduces coupled workstation interdependencies: the workload of any station directly affects the material availability of its neighbors.

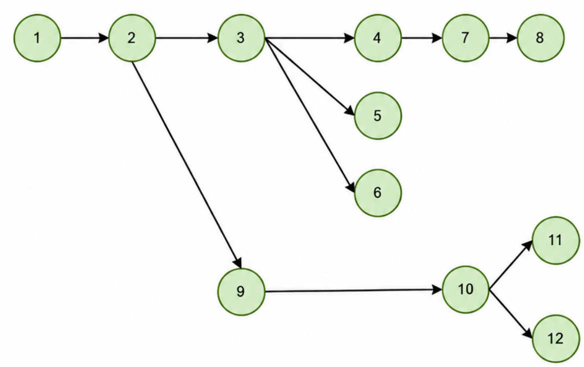
Each task $i \in I$ has a nominal disassembly time $t_i$ and yields a profit contribution $v_i$ upon completion. In manual disassembly, the actual processing time $\hat{t}_i$ deviates from $t_i$ because of operator skill variance, tool wear, and product preservation status. We assume $\hat{t}_i$ belongs to a bounded interval $[t_i, \bar{t}_i]$, where the upper bound $\bar{t}_i = t_i + \tilde{t}_i$ is determined by a deviation coefficient $p$ such that $\tilde{t}_i = p \cdot t_i$. The goal is to maximize total disassembly profit while ensuring that, under the worst-case time realizations within a budgeted uncertainty set, the cycle time constraints of all activated workstations are respected or, if violated, penalized proportionally.

Decision-making follows a two-stage structure. In the first stage, before the actual disassembly times are observed, the planner determines (i) which workstations to activate and (ii) which tasks to assign to each activated workstation. In the second stage, after the uncertain times $\hat{t}_i$ materialize, the planner observes workstation overloads and incurs delay penalties. This staged paradigm reflects industrial practice: workstation configuration and task partitioning are fixed during the planning horizon, whereas operational adjustments (overtime, penalty payments, or buffer consumption) are made reactively.

The precedence relationships among tasks are encoded by a directed acyclic graph $G = (I, E)$, where a node $i \in I$ represents a disassembly task and a directed edge $(i, j) \in E$ indicates that task $i$ must be completed before task $j$ can start. For computational storage, we define a precedence matrix $S = [s_{ij}]_{I \times I}$ as

$$s_{ij} = \begin{cases} 1, & \text{if task } i \text{ is the immediate predecessor of task } j, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Fig. 2 illustrates the precedence graph and matrix for a mobile-phone disassembly instance with $I = 12$ tasks. For example, $s_{2,3} = 1$ and $s_{2,9} = 1$ indicate that task 2 must precede tasks 3 and 9; tasks 11 and 12 both require task 10 as an immediate predecessor ($s_{10,11} = s_{10,12} = 1$).




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

Fig. 2. Mobile phone disassembly precedence graph and matrix.

To formalize the optimization model, we adopt the following assumptions:

1) Each activated workstation is assigned at least one disassembly task.
2) The total actual disassembly time of tasks assigned to workstation $w$ must not exceed the cycle time $T$; otherwise, a delay penalty proportional to the overload is incurred.

3) A workstation begins processing its first assigned task and closes upon completing its last assigned task; no idle insertion is permitted between tasks.
4) The disassembly process is profit-oriented and selective: not all tasks need to be executed.
5) The number of available workstations is fixed and finite.
6) The precedence matrix $S$ is known and constant.
7) Task profits $v_i$, nominal times $t_i$, workstation activation costs $c_w$, and unit delay penalties $C_w$ are deterministic and known.

### C. Mathematical Model

This section presents a mixed integer programming model of the DLBP, where the required notations and decision variables are defined as follows.

- Sets:

$\mathcal{W}$ Number of workstations, $\mathcal{W} = \{1, 2, \dots, W\}$, where $W$ is the number of workstations.

$\mathcal{I}$ Number of disassembly tasks, $\mathcal{I} = \{1, 2, \dots, I\}$, where $I$ is the number of disassembly tasks.

- Parameters:

$i$ Task index.
$w$ Workstation index.
$T$ Cycle time.
$\Gamma$ Fluctuation limit.
$\theta$ Delayed penalty cost (scalar parameter).
$v_i$ The disassembly income of task $i$.
$t_i$ Standard disassembly time for task $i$.
$C_w$ Unit delay penalty cost of workstation $w$.
$c_w$ Fixed start-up cost of workstation $w$.
$s_{ij}$ Precedence indicator ($s_{ij} = 1$ if task $i$ immediately precedes task $j$).
$M$ A sufficiently large positive constant.

- Decision variables:

$$x_{iw} = \begin{cases} 1, & \text{If the disassembly task } i \text{ is executed on workstation } w, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

$$u_w = \begin{cases} 1, & \text{If workstation } w \text{ is activated,} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Second-stage decision variable:
$O_w \geq 0$ The overload or delay time of workstation $w$ under uncertain task processing times. Based on these notations and decision variables, the optimization objective and constraints are formulated as follows.

*1) Deterministic objective function:* The optimization objective of this model is to achieve maximum profit, which consists of the revenue obtained from disassembly and the start-up cost of the workstation. That is,

$$f_1 = \max \sum_{i \in \mathcal{I}} \sum_{w \in \mathcal{W}} v_i x_{iw} - \sum_{w \in \mathcal{W}} c_w u_w. \tag{4}$$

*2) Model constraints:* Each task can be executed at most once:

$$\sum_{w \in \mathcal{W}} x_{iw} \leq 1, \quad \forall i \in \mathcal{I}. \tag{5}$$

Task assignment is linked to workstation activation:

$$\sum_{i \in \mathcal{I}} x_{iw} \leq M u_w, \quad \forall w \in \mathcal{W}. \tag{6}$$

The total disassembly time of each activated workstation must not exceed the cycle time:

$$\sum_{i \in \mathcal{I}} t_i x_{iw} \leq u_w T, \quad \forall w \in \mathcal{W}. \tag{7}$$

Precedence relations must be respected. If task $k$ immediately precedes task $i$ ($s_{ki} = 1$), then task $k$ must be assigned to the same or an earlier workstation than task $i$:

$$\sum_{w \in \mathcal{W}} w\,(x_{iw} - x_{kw}) + W \left( \sum_{w \in \mathcal{W}} x_{iw} - 1 \right) \leq 0, \quad \forall i \in \mathcal{I}, k \in \mathcal{I} : s_{ki} = 1. \tag{8}$$

All decision variables are binary:

$$u_w \in \{0, 1\}, \quad x_{iw} \in \{0, 1\}, \quad \forall w \in \mathcal{W}, \forall i \in \mathcal{I}. \tag{9}$$

Constraint (3) ensures that each task can only be executed once at most. Constraint (4) determines whether the workstation is activated. Constraint (5) ensures that the total task time of each workstation does not exceed the cycle time. Constraint (6) maintains the precedence order: if $s_{ki} = 1$, task $k$ must be assigned to workstation $w_k$ and task $i$ to workstation $w_i$ such that $w_k \leq w_i$; the term $W(\sum_w x_{iw} - 1)$ ensures that if task $i$ is not assigned ($\sum_w x_{iw} = 0$), the constraint is automatically satisfied. Constraint (7) specifies the binary requirements of the decision variables.

### D. Two-stage robust optimization model

*1) The first-stage objective function:* The overall objective function: The optimization objective of this model is to maximize profits. We uniformly interpret the objective function as: Total Profit = Task Revenue - Workstation Activation Cost - Worst-case Delay Penalty. Therefore, the integrated two-stage robust model is formulated as:

$$f_2 = \max_{x,u} \left( \sum_{i \in I} \sum_{w \in W} v_i x_{iw} - \sum_{w \in W} c_w u_w - Q \right) \tag{10}$$

Because solving this integrated max-max-min structure directly is intractable, we introduce an auxiliary variable $\chi$ in the master problem (detailed in Section III) to approximate the worst-case delay penalty $Q$. The first-stage decisions $(x, u)$ aim to maximize the nominal profit while minimizing $\chi$.

*2) Uncertainty set:* The budget uncertainty set is defined as

$$\mathcal{U}_t = \left\{ \hat{t} \;\middle|\; \hat{t}_i = t_i + z_i \tilde{t}_i,\ \sum_{i \in \mathcal{I}} z_i \leq \Gamma,\ 0 \leq z_i \leq 1,\ \forall i \in \mathcal{I} \right\}. \tag{11}$$

This article uses the uncertainty set $\mathcal{U}_t$ to describe the uncertain parameter $\hat{t}_i$, where $t_i$ represents the standard disassembly time of the task and $\tilde{t}_i$ represents the error time.

*3) Second-stage objective function:* 3) Second-stage objective function: The second-stage problem minimizes the delay penalty cost under the worst-case scenario. To avoid notational duplication, we drop the auxiliary symbol $\theta$ and define $Q(x, \hat{t})$ as the minimum delay penalty under a specific time realization $\hat{t}$. The worst-case recourse function $Q$ is formulated as:

$$Q = \max_{\hat{t} \in \mathcal{U}_t} \min_{O_w \geq 0} \sum_{w \in \mathcal{W}} C_w O_w \tag{12}$$

subject to

$$\sum_{i \in I} \hat{t}_i x_{iw} \leq T + O_w, \quad \forall w \in W. \tag{13}$$

Here, $Q$ represents the maximum delay penalty that may occur within the uncertainty budget set given the first-stage task assignment scheme $x$ and workstation activation scheme $u$. Among them, constraint (10) is the objective model of the second-stage problem, aiming to minimize the delay penalty cost. Constraint (11) indicates that the actual disassembly time can exceed the cycle time, and $O_w \geq 0$ ensures that no negative delay values are permitted.

## III. Algorithm Design

The two-stage robust model exhibits a max-max-min structure that standard MILP solvers cannot handle directly. We develop a tailored L-shaped decomposition that separates the first-stage integer decisions from the second-stage linear recourse. The algorithm iterates between a master problem (MP) and a subproblem (SP), generating Benders cuts to approximate the worst-case recourse function.

### A. Master Problem

At iteration $l$, the master problem optimizes the first-stage decisions using a lower-bound approximation of the second-stage cost. Let $\chi$ denote an auxiliary variable that estimates the worst-case delay penalty. The MP is formulated as:

$$\max_{x,u,\chi} \sum_{i \in I} \sum_{w \in \mathcal{W}} v_i x_{iw} - \sum_{w \in \mathcal{W}} c_w u_w - \chi \tag{14}$$

subject to

$$\chi \geq Q^k, \quad \forall k \in \{1, 2, \ldots, l-1\}, \tag{15}$$

$$\sum_{w \in \mathcal{W}} x_{iw} \leq 1, \quad \forall i \in \mathcal{I}, \tag{16}$$

$$\sum_{i \in \mathcal{I}} x_{iw} \leq M u_w, \quad \forall w \in \mathcal{W}, \tag{17}$$

$$\sum_{i \in \mathcal{I}} t_i x_{iw} \leq u_w T, \quad \forall w \in \mathcal{W}, \tag{18}$$

$$\sum_{w \in \mathcal{W}} w\,(x_{iw} - x_{kw}) + W\left(\sum_{w \in \mathcal{W}} x_{iw} - 1\right) \leq 0, \quad \forall i, k \in \mathcal{I} : s_{ki} = 1, \tag{19}$$

$$u_w \in \{0, 1\}, \quad x_{iw} \in \{0, 1\}, \quad \forall w \in \mathcal{W}, \forall i \in \mathcal{I}. \tag{20}$$

Constraint (2) represents the Benders optimality cuts: $Q^k$ is the worst-case penalty cost obtained from the subproblem at iteration $k$. Constraints (3)–(7) are the first-stage feasibility constraints. The MP yields a lower bound $LB$ on the optimal objective value.

### B. Subproblem

Given a fixed first-stage solution $(x^l, u^l)$ from the MP, the subproblem identifies the worst-case disassembly times and the corresponding minimum delay penalty. For each realization $\hat{t} \in \mathcal{U}_t$, the inner recourse problem minimizes the total penalty:

$$Q(x^l, \hat{t}) = \min_{O_w \geq 0} \sum_{w \in \mathcal{W}} C_w O_w \tag{21}$$

subject to

$$\sum_{i \in \mathcal{I}} \hat{t}_i x_{iw}^l \leq T + O_w, \quad \forall w \in \mathcal{W}. \tag{22}$$

The subproblem then maximizes this minimum penalty over the uncertainty set:

$$Q^l = \max_{\hat{t} \in \mathcal{U}_t} Q(x^l, \hat{t}). \tag{23}$$

To solve this max-min bilevel problem, we apply strong duality to the inner minimization. Associating dual variables $\lambda_w \geq 0$ with constraints (9), the dual of the inner problem is:

$$\max_{\lambda_w \geq 0} \sum_{w \in \mathcal{W}} \lambda_w \left(T - \sum_{i \in \mathcal{I}} \hat{t}_i x_{iw}^l\right) \tag{24}$$

subject to

$$\lambda_w \leq C_w, \quad \forall w \in \mathcal{W}. \tag{25}$$

Substituting this dual into the outer maximization, the subproblem becomes a single-level mixed-integer linear program:

$$Q^l = \max_{\hat{t}, \lambda} \sum_{w \in \mathcal{W}} \lambda_w \left(T - \sum_{i \in \mathcal{I}} \hat{t}_i x_{iw}^l\right) \tag{26}$$

subject to

$$\hat{t} \in \mathcal{U}_t, \tag{27}$$

$$0 \leq \lambda_w \leq C_w, \quad \forall w \in \mathcal{W}. \tag{28}$$

The optimal value $Q^l$ provides an upper bound on the true recourse cost for the current first-stage solution $(x^l, u^l)$. The optimality cut generation mechanism works as follows: we evaluate the gap between the upper bound $Q^l$ (from the SP) and the lower bound estimation $\chi^l$ (from the MP). If $Q^l - \chi^l > \epsilon$, where $\epsilon$ is a predefined small tolerance, it indicates that the master problem has underestimated the worst-case delay penalty. Consequently, the optimality cut $\chi \geq Q^l$, parameterized by the worst-case realization $\hat{t}^l$ identified in the current iteration, is generated and appended to the master problem to refine its feasible region. If $Q^l - \chi^l \leq \epsilon$, the algorithm terminates, confirming that the current first-stage decisions are robust against all scenarios within the uncertainty set.

To rigorously establish the validity of this procedure, we present the convergence proof of the proposed L-shaped decomposition.

**Theorem 1 (Finite Convergence).** *The proposed L-shaped decomposition algorithm converges to the global optimal solution in a finite number of iterations.*

*Proof.* The convergence relies on the structural properties of the budget uncertainty set $\mathcal{U}_t$ and the recourse function. First, the budget uncertainty set $\mathcal{U}_t$ defined in this problem is a bounded, closed polyhedron. According to the fundamental theorem of linear programming, the maximum of a convex recourse function over a polyhedral set occurs at one of its extreme points. Let $\mathcal{E}(\mathcal{U}_t)$ denote the set of all extreme points of $\mathcal{U}_t$. Since $\mathcal{U}_t$ is defined by a finite number of linear constraints (variable bounds and the cardinality budget $\Gamma$), the cardinality of $\mathcal{E}(\mathcal{U}_t)$, denoted as $|\mathcal{E}(\mathcal{U}_t)|$, is strictly finite.

Second, at each iteration $l$, if the current solution is not optimal ($Q^l > \chi^l$), the subproblem identifies a specific worst-case scenario $\hat{t}^l \in \mathcal{E}(\mathcal{U}_t)$ and generates a Benders optimality cut $\chi \geq Q^l$. Because this cut strictly isolates the current suboptimal first-stage estimation $(x^l, u^l, \chi^l)$ from the feasible region without eliminating any valid optimal solutions, no extreme point $\hat{t}^l$ can be generated more than once.

Therefore, in the absolute worst-case scenario, the algorithm will implicitly enumerate all $|\mathcal{E}(\mathcal{U}_t)|$ extreme points. Since $|\mathcal{E}(\mathcal{U}_t)| < \infty$, the master problem will collect a finite number of valid cuts, fully characterizing the true worst-case penalty function. The gap between the upper bound and lower bound is monotonically non-increasing and must eventually satisfy the condition $UB - LB \leq \epsilon$ in a finite number of steps, thereby guaranteeing global optimality upon termination.

### C. Algorithm Flow

The L-shaped algorithm proceeds as follows.

**Step 1: Initialization.** Set iteration counter $l = 1$, lower bound $LB = -\infty$, upper bound $UB = +\infty$, and initialize the cut set $Q = \emptyset$.

**Step 2: Solve MP.** Solve the master problem (1)–(7) to obtain solution $(x^l, u^l, \chi^l)$. Update the lower bound:

$$LB = \sum_{i \in \mathcal{I}} \sum_{w \in \mathcal{W}} v_i x_{iw}^l - \sum_{w \in \mathcal{W}} c_w u_w^l - \chi^l. \tag{29}$$

**Step 3: Solve SP.** Solve the subproblem (13)–(15) with fixed $(x^l, u^l)$ to obtain the worst-case scenario $\hat{t}^l$ and optimal value $Q^l$. Update the upper bound:

$$UB = \sum_{i \in \mathcal{I}} \sum_{w \in \mathcal{W}} v_i x_{iw}^l - \sum_{w \in \mathcal{W}} c_w u_w^l - Q^l. \tag{30}$$

**Step 4: Check convergence.** If $UB - LB \leq \varepsilon$ or $l \geq L_{\max}$, stop and return $(x^l, u^l)$ as the optimal solution. Otherwise, add the optimality cut $\chi \geq Q^l$ to $Q$, set $l = l + 1$, and return to Step 2.

**Algorithm 1** L-shaped Algorithm for Two-Stage Robust Optimization

**Input:** Task profits $v_i$, nominal times $t_i$, deviation $\tilde{t}_i$, cycle time $T$, uncertainty budget $\Gamma$, workstation costs $c_w$, penalty coefficients $C_w$, precedence matrix $S$, convergence tolerance $\varepsilon$, maximum iterations $L_{\max}$
**Output:** Optimal first-stage solution $(x^*, u^*)$ and optimal objective value $f^*$
Initialize $l \leftarrow 1$, $LB \leftarrow -\infty$, $UB \leftarrow +\infty$, $Q \leftarrow \emptyset$
**Step 1: Solve Initial Master Problem**
Solve MP (1)–(7) without cuts to obtain $(x^1, u^1, \chi^1)$
Update $LB \leftarrow \sum_{i,w} v_i x_{iw}^1 - \sum_w c_w u_w^1 - \chi^1$
**while** $UB - LB > \varepsilon$ and $l < L_{\max}$ **do**
  **Step 2: Solve Subproblem (Upper Bound Update)**
  Fix $x = x^l$, solve SP (13)–(15) to obtain $Q^l$ and worst-case $\hat{t}^l$
  Update $UB \leftarrow \sum_{i,w} v_i x_{iw}^l - \sum_w c_w u_w^l - Q^l$
  **Step 3: Generate Optimality Cut**
  Add cut $\chi \geq Q^l$ to $Q$
  **Step 4: Solve Updated Master Problem (Lower Bound Update)**
  Solve MP (1)–(7) with all cuts in $Q$ to obtain $(x^{l+1}, u^{l+1}, \chi^{l+1})$
  Update $LB \leftarrow \sum_{i,w} v_i x_{iw}^{l+1} - \sum_w c_w u_w^{l+1} - \chi^{l+1}$
  $l \leftarrow l + 1$
**end while**
Return $(x^*, u^*) \leftarrow (x^l, u^l)$ and $f^* \leftarrow UB$

### D. Algorithm Analysis

The L-shaped algorithm offers both theoretical tractability and practical efficiency for the proposed two-stage robust model. At the theoretical level, the Benders cut generation mechanism dynamically refines the master problem by adding only those cuts that are binding at the current solution, avoiding the static enumeration of all uncertainty scenarios. This row-generation strategy reduces the effective problem size compared with full scenario enumeration, particularly when the uncertainty budget $\Gamma$ is small relative to the total number of tasks $I$.

At the computational level, the decomposition exploits the problem structure: the MP contains only binary first-stage variables and a single continuous auxiliary variable $\chi$, while the SP reduces to a mixed-integer linear program of modest size after strong-duality transformation. The synergy between row generation (cut addition) and bound tightening ($LB$/$UB$ update) ensures finite convergence, because the number of distinct extreme points of the uncertainty set $\mathcal{U}_t$ is finite for bounded $\Gamma$.

In terms of industrial applicability, the algorithm design aligns with the operational reality of disassembly scheduling. The hierarchical architecture decouples strategic decisions (workstation activation and task assignment) from operational adjustments (delay penalty after observing actual task times), supporting collaborative optimization of resource allocation and risk management. The model focuses on the aggregate completion of disassembly tasks within each cycle, abstracting away low-level details such as individual worker assignments

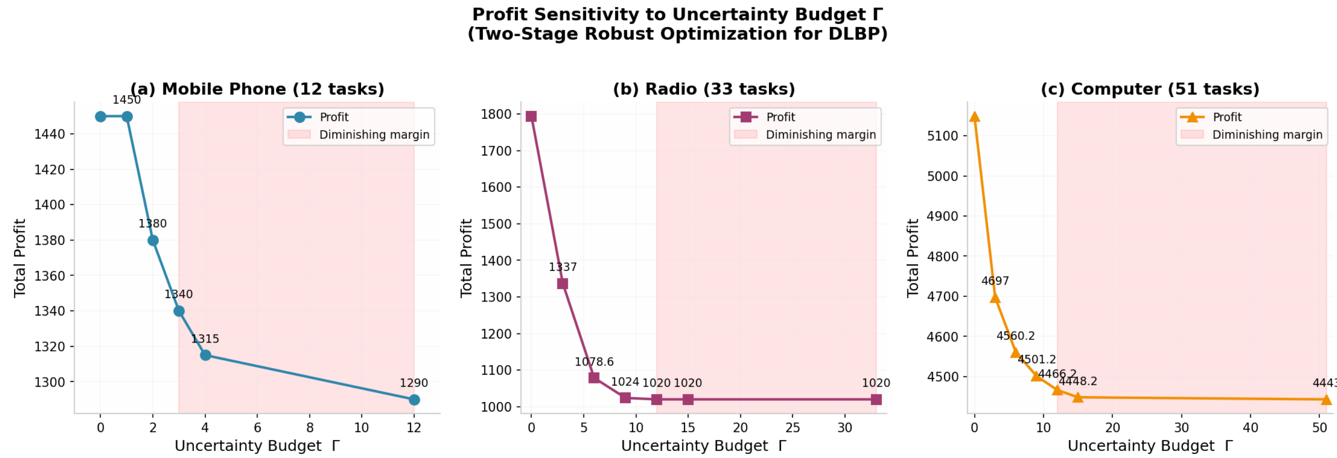



Fig. 3. Profit sensitivity to uncertainty budget Γ.

or specific tool sequences that have marginal impact on overall profit but significantly complicate the optimization. This abstraction level is appropriate for tactical planning, where the primary concern is whether tasks are completed within the cycle time of activated workstations rather than the microscopic execution details.

## IV. Experimental Design

To evaluate the model accuracy and algorithmic efficiency, the proposed L-shaped decomposition was implemented in Python 3.8 using PyCharm 2020, and executed on a Windows 10 PC with an AMD 5800H CPU (3.20 GHz) and 24 GB RAM.

### A. Experimental Setup

Three benchmark instances of increasing scale were adopted from the disassembly line balancing literature [31]: a mobile phone with 12 tasks, a radio with 33 tasks, and a computer with 51 tasks. Table I summarizes the task characteristics, and Table II lists the workstation configurations.

TABLE I Product Information

| Product | Tasks | Time | Profit |
|---|---|---|---|
| Mobile phone | 12 | 5–9 | 100–150 |
| Radio | 33 | 5–77 | 15–100 |
| Computer | 51 | 1–75 | 13–2229 |

TABLE II Workstation Information

| Product | WS | Act. cost | Pen. cost |
|---|---|---|---|
| Mobile phone | 4 | {10,12,13,15} | {25,35,50,40} |
| Radio | 5 | {10,12,13,15,18} | {25,35,50,40,45} |
| Computer | 5 | {10,12,13,15,150} | {25,35,50,40,45} |

The precedence relationships for the mobile phone instance are illustrated in Fig. 2 (see Section II). The two-stage iterative structure of the L-shaped algorithm is shown in Fig. **??**. For each instance, the uncertainty budget Γ was varied to examine the profit–robustness trade-off.

### B. Computational Results

Tables III–V report the optimal profit, iteration count, and CPU time for each case.

TABLE III Mobile phone results ($I$ = 12, $W$ = 4)

| Γ | Profit | Iter. | Time (s) |
|---|---|---|---|
| 0 | 1450 | 1 | 0.19 |
| 1 | 1450 | 1 | 0.18 |
| 2 | 1380 | 3 | 1.58 |
| 3 | 1340 | 3 | 1.15 |
| 4 | 1315 | 3 | 1.54 |
| 12 | 1290 | 3 | 1.16 |

TABLE IV Radio results ($I$ = 33, $W$ = 5)

| Γ | Profit | Iter. | Time (s) |
|---|---|---|---|
| 0 | 1794 | 1 | 0.46 |
| 3 | 1337 | 3 | 0.96 |
| 6 | 1078.6 | 4 | 1.49 |
| 9 | 1024 | 3 | 0.95 |
| 12 | 1020 | 3 | 0.96 |
| 15 | 1020 | 3 | 0.94 |
| 33 | 1020 | 2 | 0.97 |

TABLE V Computer results ($I$ = 51, $W$ = 5)

| Γ | Profit | Iter. | Time (s) |
|---|---|---|---|
| 0 | 5149 | 1 | 0.22 |
| 3 | 4697 | 2 | 0.47 |
| 6 | 4560.2 | 2 | 1.44 |
| 9 | 4501.2 | 3 | 2.15 |
| 12 | 4466.2 | 2 | 0.44 |
| 15 | 4448.2 | 2 | 1.16 |
| 51 | 4443 | 2 | 0.47 |

### C. Experimental Analysis

The analysis of experimental results indicates that the L-shaped algorithm exhibits good solving performance and stability across cases of different scales. The specific analysis is as follows.

All three instances exhibit rapid convergence, with the number of iterations generally maintained between 1 and 4, indicating good convergence efficiency. As the uncertainty budget $\Gamma$ increases, the number of iterations does not significantly increase, indicating that the algorithm has strong robustness to changes in uncertain parameters.

In terms of computation time, the solving time for all cases was controlled within 3 seconds, and even for the large-scale computer instance (51 tasks), the maximum computation time was only 2.15 seconds. The computation time is positively correlated with the number of iterations, but not strictly linearly, indicating that the computational complexity of the algorithm varies at different stages of iteration.

With the increase of the uncertainty budget $\Gamma$, the profit exhibits a notable downward trend, indicating that the algorithm can effectively utilize uncertain information to optimize decision-making. In the small-scale instance (mobile phone), the profit decreases from 1450 ($\Gamma$ = 0) to 1290 ($\Gamma$ = 12), representing a reduction of 11.0%. In the medium-scale instance (radio), the profit drops from 1794 ($\Gamma$ = 0) to 1020 ($\Gamma$ = 33), with a reduction of 43.1%. In the large-scale instance (computer), the profit reduces from 5149 ($\Gamma$ = 0) to 4443 ($\Gamma$ = 51), achieving a reduction of 13.7%. The increase in $\Gamma$ has a marginal diminishing effect on the profit, which is particularly evident in medium-to-large-scale instances. In the radio instance, when $\Gamma$ increases from 9 to 33, the profit only decreases from 1024 to 1020, indicating that the effect tends to saturate. In the computer instance, when $\Gamma$ increases from 15 to 51, the profit only decreases from 4448.2 to 4443, also showing a convergence trend.

As the scale of the problem increases, the initial profit (with $\Gamma$ = 0) increases significantly, but the growth rate of the optimized profit is relatively low. The medium-scale instance (radio) exhibits the highest reduction amplitude (43.1%), possibly because its moderate scale provides sufficient optimization space without limiting the effect due to excessive scale.

The L-shaped algorithm excels in handling disassembly scheduling problems of various scales, demonstrating not only high computational efficiency but also effective utilization of uncertainty information for decision optimization. Notably, the algorithm exhibits optimal performance on medium-scale problems, providing a reference for the reasonable selection of problem scales in practical applications. The marginal diminishing effect of $\Gamma$ also suggests that, in practical applications, a balance needs to be struck between the granularity of uncertainty modeling and computational costs, selecting appropriate levels of uncertainty budget.

## V. Conclusion

This paper establishes a two-stage robust optimization model for circular disassembly line balancing under disassembly time uncertainty, with the objective of maximizing disassembly profit. The effectiveness of the model was verified through theoretical analysis and numerical experiments [32]. In-depth analysis was conducted on the variation patterns of disassembly plans under different levels of uncertainty budget and product types. The experimental results indicate that as the uncertainty budget $\Gamma$ increases, the profit of the disassembly line exhibits a downward trend—specifically demonstrating maximum profit reduction rates of 11.0%, 43.1%, and 13.7% for the small, medium, and large-scale instances, respectively—verifying the negative impact of time uncertainty on disassembly performance. Furthermore, the tailored L-shaped decomposition proved highly efficient, consistently converging within 4 iterations with a maximum solving time of only 2.15 seconds.

For circular disassembly line balancing with uncertain disassembly time, the two-stage robust optimization method decomposes the problem into a deterministic master problem and a recourse subproblem. The first-stage decision determines the basic disassembly plan (task assignment and workstation activation), while the second-stage decision adjusts the delay penalty strategy based on the actual disassembly time, effectively balancing disassembly profit and delay risk [33]. By adjusting the uncertainty budget $\Gamma$, a series of robust solutions can be provided to decision-makers to meet the needs of different risk preferences. This method provides more choices for enterprise decision-makers, enabling them to improve disassembly profits or reduce production line delay probability based on actual situations, thereby enhancing disassembly line efficiency and overall competitiveness [34].

Future work includes incorporating additional practical factors into the optimization model, such as carbon emissions, uncertain probability of successful disassembly, and machine wear. In addition, we plan to introduce distributed robust optimization methods to better handle distributional information of uncertain parameters, and adopt column-and-constraint generation (CCG) algorithms to improve the efficiency of solving large-scale problems. We will also explore the combination of reinforcement learning and other advanced intelligent computing algorithms to solve complex disassembly line optimization problems, further enhancing the practicality and computational performance of the model [35].

## References


[1] Y. Ji, S. Liu, M. Zhou, Z. Zhao, X. Guo, and L. Qi, "A machine learning and genetic algorithm-based method for predicting width deviation of hot-rolled strip in steel production systems," *Information Sciences*, vol. 589, pp. 360–375, 2022.

[2] Z. Zhao, S. Liu, M. Zhou, D. You, and X. Guo, "Heuristic scheduling of batch production processes based on petri nets and iterated greedy algorithms," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 1, pp. 251–261, 2022.

[3] 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, 2022.

[4] Z. Zhao, S. Liu, M. Zhou, and A. Abusorrah, "Dual-objective mixed integer linear program and memetic algorithm for an industrial group scheduling problem," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 6, pp. 1199–1209, 2022.

[5] M. Geissdoerfer, P. Savaget, N. M. Bocken, and E. J. Hultink, "The circular economy – a new sustainability paradigm?" *Journal of Cleaner Production*, vol. 143, pp. 757–768, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959652616321023

[6] 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.

[7] 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.

[8] L. Qi, W. Luan, X. S. Lu, and X. Guo, "Shared p-type logic petri net composition and property analysis: A vector computational method," *IEEE Access*, vol. 8, pp. 34 644–34 653, 2020.

[9] K. Zhang, Y. Su, X. Guo, L. Qi, and Z. Zhao, "Mu-gan: Facial attribute editing based on multi-attention mechanism," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 9, pp. 1614–1626, 2020.

[10] 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.

[11] Z. Bi, X. Guo, J. Wang, S. Qin, and G. Liu, "Deep reinforcement learning for truck-drone delivery problem," *Drones*, vol. 7, no. 7, p. 445, 2023.

[12] X. Guo, Y. Feng, J. Wang, S. Qin, L. Qi, B. Hu, and Y. Ji, "Solving human–robot collaborative circular disassembly line balancing problem via graph neural network-enhanced proximal policy optimization algorithm," *IEEE Transactions on Computational Social Systems*, vol. 13, no. 3, pp. 3476–3490, 2026.

[13] S. Qin, F. Guo, J. Wang, L. Qi, J. Wang, X. Guo, and Z. Zhao, "Expanded discrete migratory bird optimizer for circular disassembly line balancing with tool deterioration and replacement," *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 1, 2025.

[14] C. Xu, H. Wei, X. Guo, S. Liu, L. Qi, and Z. Zhao, "Human-robot collaboration multi-objective disassembly line balancing subject to task failure via multi-objective artificial bee colony algorithm," *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 1–6, 2020.

[15] W. Xu, J. Cui, B. Liu, J. Liu, B. Yao, and Z. Zhou, "Human-robot collaborative disassembly line balancing considering the safe strategy in remanufacturing," *Journal of Cleaner Production*, vol. 324, p. 129158, 2021.

[16] J. Wang, M. Zhou, X. Guo, and L. Qi, "Multiperiod asset allocation considering dynamic loss aversion behavior of investors," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 1, pp. 73–81, 2018.

[17] S. Qin, S. Zhang, J. Wang, S. Liu, X. Guo, and L. Qi, "Multi-objective multi-verse optimizer for multi-robotic u-shaped disassembly line balancing problems," *IEEE Transactions on Artificial Intelligence*, 2023.

[18] 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.

[19] 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*, pp. 1–14, 2023.

[20] 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.

[21] Y. Fu, M. Zhou, X. Guo, and L. Qi, "Scheduling dual-objective stochastic hybrid flow shop with deteriorating jobs via bi-population evolutionary algorithm," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 12, pp. 5037–5048, 2019.

[22] L. Huang, M. Zhou, K. Hao, and E. Hou, "A survey of multi-robot regular and adversarial patrolling," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 4, pp. 894–903, 2019.

[23] H. Zhang, D. Pham, and Q. Kang, "Improved fruit fly algorithm for multi-objective disassembly line balancing problem considering learning effect," *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 2, pp. 108–118, June 2025.

[24] Y. Tan, M. Zhou, Y. Zhang, X. Guo, L. Qi, and Y. Wang, "Hybrid scatter search algorithm for optimal and energy-efficient steelmaking-continuous casting," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1814–1828, 2020.

[25] 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.

[26] 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.

[27] 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, 2022.

[28] L. Qi, L. Liang, W. Luan, T. Lu, X. Guo, and Q. T. A. Talukder, "A novel framework combining vsl and vehicle platooning for freeway bottleneck," *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 1, pp. 47–57, March 2025.

[29] Y. Tan, M. Zhou, Y. Wang, X. Guo, and L. Qi, "A hybrid mip–cp approach to multistage scheduling problem in continuous casting and hot-rolling processes," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1860–1869, 2019.

[30] L. Zhou, H. Zhu, and B. Akbari, "Multi-objective optimization of multi-factory remanufacturing process considering worker fatigue," *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 2, pp. 93–107, June 2025.

[31] 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, 2020.

[32] J. Gu, J. Wang, X. Guo, G. Liu, S. Qin, and Z. Bi, "A metaverse-based teaching building evacuation training system with deep reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 4, pp. 2209–2219, 2023.

[33] N. Yang, Z. Zheng, M. Zhou, and et al., "A domain-guided noise-optimization-based inversion method for facial image manipulation," *IEEE Transactions on Image Processing*, vol. 30, pp. 6198–6211, 2021.

[34] Z. Zhao, S. Liu, M. Zhou, D. You, and X. Guo, "Heuristic scheduling of batch production processes based on petri nets and iterated greedy algorithms," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 1, pp. 251–261, 2022.

[35] S. Dai, Z. Zhang, W. Wang, C. Li, J. Parron, and E. Herrera, "Human-robot collaborative disassembly profit maximization via improved grey wolf optimizer," *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 2, pp. 12–22, 2025.

**Kun Wang** received the B.S. degree in information and computing science from Zhejiang University of Finance and Economics, Hangzhou, China, in 2019. He is currently a graduate student with the School of Artificial Intelligence and Software, Liaoning Petrochemical University, Fushun, China. His research interests include artificial intelligence and intelligent optimization algorithms.

**Zihao Wei** received his degree in computer science and technology from Anyang Institute of Technology in China, in 2023. Now he is a graduate student of College of Artificial Intelligence and Software,Liaoning University of Petrochemical Technology. His main research direction is reinforcement learning.