LLM-Assisted Warm-Start Optimization for Multi-Factory Human-Robot Collaborative Disassembly Scheduling

Minghui Shi and Jialong Li

***Abstract*—The quality of initial solutions strongly affects the convergence behavior of meta-heuristic algorithms for human-robot collaborative disassembly scheduling. This paper proposes a Large Language Model (LLM)-assisted warm-start method for generating feasible task sequences in a multi-factory human-robot collaborative disassembly problem. A structured instruction set is designed to describe product disassembly information and task constraints, including precedence and conflict relationships derived from AND/OR graphs. The generated sequences are parsed, validated, and injected into the initial population of an Intelligent Discrete Optimization Algorithm (IDOA). A profit-oriented mathematical model is used as the evaluation basis for task-tool assignment, workstation allocation, and disassembler configuration. Numerical experiments on several end-of-life (EoL) products show that the proposed LLM-assisted initialization method improves initial solution quality and accelerates convergence in terms of iteration count. Although local LLM inference introduces additional runtime overhead, the results demonstrate the feasibility of using LLMs as warm-start generators for combinatorial disassembly optimization.**

***Index Terms*—Human-Robot Collaborative Disassembly Line; Multi-Factory Optimization; Resource Configuration; Large Language Model (LLM); Meta-heuristic Algorithm**

I. Introduction

END-OF-LIFE (EoL) product disassembly plays an important role in remanufacturing and resource recovery. With the increasing variety and structural complexity of EoL products, disassembly systems require specialized equipment, diverse tools, and systematic task planning [1, 2, 3]. In human-robot collaborative disassembly lines, tasks must be assigned to workstations, disassemblers, and compatible tools while satisfying precedence and conflict constraints. Human operators provide flexibility for uncertain and complex operations, while robots improve execution stability and efficiency. However, this collaboration also introduces additional assignment, balancing, and resource configuration decisions [4, 5, 6].

In multi-factory disassembly environments, the optimization problem becomes more complicated because factories usually differ in machinery availability, tool configurations, workstation opening decisions, and disassembler allocation. These heterogeneous resource capacities directly affect disassembly efficiency, product-to-factory assignment, and total recovery profit [7, 8, 9]. Existing profit-oriented and resource-constrained disassembly studies provide useful modeling foundations for such problems [4, 9, 10]. Nevertheless, exact solvers may become inefficient as the problem scale increases, and meta-heuristic algorithms are often adopted to obtain high-quality solutions within acceptable computational time.

Population-based meta-heuristic algorithms are sensitive to the quality of initial solutions. Randomly generated task sequences may violate precedence or conflict constraints defined by AND/OR graphs, and poor initial populations may slow down convergence or lead the search to low-quality regions. Therefore, generating feasible and high-quality initial task sequences is important for improving the practical performance of meta-heuristic optimization in disassembly scheduling. Recent studies have shown that Large Language Models (LLMs) can serve as candidate-solution generators, optimizers, or hyper-heuristic components in combinatorial optimization and evolutionary search [11, 12, 13]. These studies indicate that LLMs have potential to use structured problem descriptions and historical solution patterns to generate useful candidate solutions.

Motivated by this observation, this paper proposes an LLM-assisted warm-start strategy for the Intelligent Discrete Optimization Algorithm (IDOA). Instead of relying only on random initialization, the proposed method constructs structured instructions from product disassembly information and task constraints. The LLM then generates candidate task sequences, which are parsed, validated, and inserted into the initial IDOA population. The main purpose of this design is not to replace the meta-heuristic search process, but to provide a more informative starting population for subsequent optimization.[14, 15, 16, 17, 18, 19, 20]

The main contributions of this paper are summarized as follows. First, an LLM-assisted initial solution generation method is developed for human-robot collaborative disassembly scheduling. Second, a structured instruction mechanism is designed to guide the LLM in generating task sequences under precedence and conflict constraints. Third, the LLM-generated sequences are integrated into IDOA through a warm-start population initialization strategy. Finally, numerical experiments on multiple EoL product cases are conducted to evaluate the effect of LLM-assisted initialization on initial solution quality and convergence behavior.

The remainder of this paper is organized as follows. Section II describes the considered multi-factory human-robot collaborative disassembly problem. Section III presents the

M. Shi and J. Li are with the Artificial Intelligence and Software College, Liaoning Petrochemical University, Fushun 113001, P. R. China, (e-mail: Fnex1024s@163.com, lijialong@stu.lnpu.edu.cn).

Corresponding author: Minghui Shi.

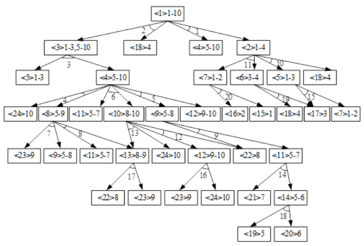

Fig. 1. The AND/OR graph of the car battery

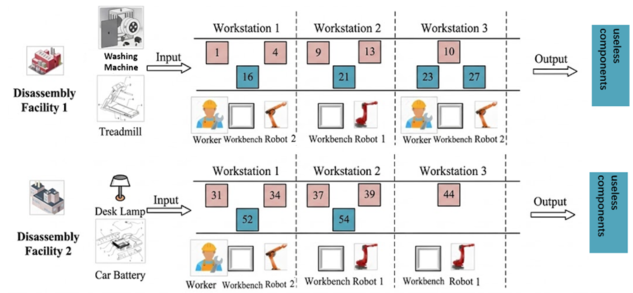

Fig. 2. Multi-factory human-robot collaborative disassembly line

profit-oriented mathematical model. Section IV introduces the LLM-assisted warm-start method. Section V describes the IDOA-based optimization procedure. Section VI reports the experimental results. Section VII concludes this paper.

II. Problem Description

A. Problem Characteristics

This paper considers a multi-factory human-robot collaborative disassembly scheduling problem. Each EoL product contains a set of disassembly tasks, and each task must be assigned to a disassembly factory, a workstation, a disassembler, and a compatible tool. The disassembler can be a human operator, a robot, or a human-robot collaborative unit. The assignment must satisfy resource availability, tool compatibility, and task precedence/conflict constraints [4, 8, 9].

In the considered multi-factory environment, factories differ in tool types, tool quantities, workstation configurations, and disassembler availability. These heterogeneous resources directly affect task allocation and total recovery profit. Therefore, the problem involves both scheduling decisions and resource configuration decisions. The objective is to maximize the total profit obtained from recovered subassemblies after subtracting task execution costs, transportation costs, workstation/factory opening costs, and tool utilization costs [9, 10, 21].

The focus of this paper is not to redesign the complete disassembly optimization model, but to improve the initialization stage of the meta-heuristic solver. Randomly generated task sequences may violate precedence or conflict constraints and may provide poor starting points for search. To address this issue, an LLM is employed to generate feasible or near-feasible initial task sequences for EoL products. These sequences are then parsed, validated, and used as warm-start solutions for IDOA [11, 12, 13].

B. Case Description

In this study, desk lamps, washing machines, treadmills, and car batteries are selected as representative EoL products. Their disassembly processes are described by AND/OR graphs, which are commonly used to represent alternative disassembly paths, precedence relationships, and conflict constraints among tasks [1, 3, 22, 23].

Fig. 1 shows the AND/OR graph of a car battery. In this graph, nodes represent subassemblies or components, while directed links describe feasible disassembly operations and their logical relationships. The precedence and conflict information extracted from the AND/OR graph is used not only in the mathematical model, but also in the LLM-assisted sequence generation process.

C. Conceptual Model

Fig. 2 shows the considered multi-factory human-robot collaborative disassembly line. Each factory contains several workstations, and each workstation may be equipped with different tool types. During scheduling, each task is assigned to a factory and a workstation with compatible tools. The task is then executed by a human operator, a robot, or a human-robot collaborative unit according to the resource configuration.

The AND/OR graph is converted into three matrices to support constraint checking and sequence validation. The disassembly matrix describes the generation and consumption relationships between tasks and subassemblies. The precedence matrix records the sequential constraints between tasks. The conflict matrix records mutually exclusive task relationships. These matrices provide the structured constraint information used by both IDOA and the proposed LLM-assisted warm-start mechanism [1, 3, 23].

$$d_{pij} = \begin{cases} 1, & \text{if task } j \text{ generates subassembly } i \\ -1, & \text{if task } j \text{ consumes subassembly } i \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$s_{pj_1j_2} = \begin{cases} 1, & \text{if task } j_1 \text{ precedes task } j_2 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$r_{pj_1j_2} = \begin{cases} 1, & \text{if tasks } j_1 \text{ and } j_2 \text{ conflict} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where d_{pij} denotes the relationship between task j and subassembly i of product p, $s_{pj_1j_2}$ denotes the precedence relationship between tasks j_1 and j_2, and $r_{pj_1j_2}$ denotes the conflict relationship between tasks j_1 and j_2.

III. Mathematical Model

This section presents a compact profit-oriented model used as the fitness evaluation basis of IDOA and the proposed LLM-assisted warm-start strategy. The model evaluates each decoded solution according to recovered subassembly revenue, task execution cost, transportation cost, tool utilization cost, factory operating cost, and workstation opening cost.

The following assumptions are adopted. First, the processing costs of human operators and robots are deterministic and known. Second, the recovery prices of subassemblies are known. Third, the operating costs of factories and the opening costs of workstations are known. Fourth, the precedence and conflict relationships among tasks are predefined by the AND/OR graph. Fifth, each task must be assigned to a compatible tool. Finally, the wear-and-tear cost of each tool used for each disassembly task is known.

Let $p \in P$ denote products, $i \in I$ denote subassemblies, $j \in J$ denote disassembly tasks, $k \in K$ denote disassembly factories, $w \in W$ denote workstations, $m \in M$ denote remanufacturing plants, $n \in N$ denote robot types, $h \in H$ denote human workers, and $r \in R$ denote tool types. The binary variable z_{pk} indicates whether product p is assigned to factory k. The variable x_{pjkw} indicates whether task j of product p is assigned to workstation w in factory k. The variables x_{pjkwn} and x_{pjkwh} indicate whether the task is executed by robot n or human worker h, respectively. The variable Z_{pjkwr} indicates whether task j of product p uses tool type r at workstation w in factory k. The variable α_{kmpi} indicates whether subassembly i of product p is transported from factory k to remanufacturing plant m. The variable T_k denotes the cycle time of factory k.

The objective function maximizes the total recovery profit. It subtracts human execution cost, robot execution cost, tool utilization cost, factory operating cost, and workstation opening cost from recovered subassembly revenues [8, 9, 10, 21].

$$\begin{aligned}\max f = & \sum_{k\in K}\sum_{m\in M}\sum_{p\in P}\sum_{i\in I}(v_{mpi}-c_{kmpi})\alpha_{kmpi} \\ & - \sum_{k\in K}\sum_{p\in P}\sum_{j\in J}\sum_{w\in W}\sum_{h\in H} h_{pj}H_{pj}x_{pjkwh} \\ & - \sum_{k\in K}\sum_{p\in P}\sum_{j\in J}\sum_{w\in W}\sum_{n\in N} r_{pnj}R_{pnj}x_{pjkwn} \\ & - \sum_{k\in K}\sum_{p\in P}\sum_{j\in J}\sum_{w\in W}\sum_{r\in R} Z_{pjkwr}C_{pjr} \\ & - \sum_{k\in K} c_k T_k - \sum_{k\in K}\sum_{w\in W} c_{kw}u_{kw}.\end{aligned} \tag{4}$$

Several key constraints are retained to show the main feasibility logic of the model. Each product is assigned to one factory:

$$\sum_{k\in K} z_{pk} = 1, \quad \forall p \in P. \tag{5}$$

A task can be assigned only to the factory selected for its product and to an opened workstation:

$$\sum_{w\in W} x_{pjkw} \le z_{pk}, \quad \forall p \in P, \forall j \in J, \forall k \in K, \tag{6}$$

$$x_{pjkw} \le u_{kw}, \quad \forall p \in P, \forall j \in J, \forall k \in K, \forall w \in W. \tag{7}$$

The execution variables are linked with the task assignment variable:

$$\sum_{n\in N} x_{pjkwn} + \sum_{h\in H} x_{pjkwh} = x_{pjkw}, \quad \forall p \in P, \forall j \in J, \forall k \in K, \forall w \in W. \tag{8}$$

The workload of each workstation cannot exceed the cycle time of its factory:

$$\sum_{p\in P}\sum_{j\in J}\left(\sum_{n\in N} r_{pnj}x_{pjkwn} + \sum_{h\in H} h_{pj}x_{pjkwh}\right) \le T_k, \quad \forall k \in K, \forall w \in W. \tag{9}$$

Precedence and conflict relationships are derived from the AND/OR graph. If task j_1 precedes task j_2, then j_1 should be assigned no later than j_2:

$$\sum_{w\in W} w x_{pj_1kw} \le \sum_{w\in W} w x_{pj_2kw}, \quad \forall p \in P, \forall k \in K, s_{pj_1j_2} = 1. \tag{10}$$

Mutually exclusive tasks cannot be selected simultaneously:

$$\sum_{w\in W}\left(x_{pj_1kw} + x_{pj_2kw}\right) \le 1, \quad \forall p \in P, \forall k \in K, r_{pj_1j_2} = 1. \tag{11}$$

Tool compatibility is enforced by linking task assignment, task-tool requirements, and workstation-tool availability:

$$Z_{pjkwr} \le x_{pjkw}, \quad \forall p, j, k, w, r, \tag{12}$$

$$Z_{pjkwr} \le U_{pjr}, \quad \forall p, j, k, w, r, \tag{13}$$

$$Z_{pjkwr} \le U_{kwr}, \quad \forall p, j, k, w, r, \tag{14}$$

$$x_{pjkw} \le \sum_{r\in R} Z_{pjkwr}, \quad \forall p \in P, \forall j \in J, \forall k \in K, \forall w \in W. \tag{15}$$

These constraints ensure that each decoded individual corresponds to a feasible multi-factory human-robot collaborative disassembly schedule. Other logical constraints, such as factory activation and resource-domain restrictions, are implemented in the decoding and feasibility-checking procedure.

IV. Proposed LLM-Warm-Started IDOA

This section presents the proposed LLM-warm-started IDOA. The key idea is to use the LLM as an initial solution generator. The LLM generates candidate task sequences from structured disassembly instructions, and the validated sequences are inserted into the initial population of IDOA. The main search operators of IDOA remain unchanged.

A. Structured Instruction Design

Recent studies show that LLMs can serve as optimizers, heuristic generators, or hyper-heuristic components for combinatorial optimization and evolutionary search [11, 12, 13]. Inspired by these studies, this paper constructs a structured instruction to guide the LLM in generating feasible or near-feasible disassembly task sequences.

Each instruction contains three parts: task description, input information, and output format. The task description specifies the product and sequence generation requirement. The input information contains candidate task relations and disassembly constraints. The output format restricts the LLM response to a list of task indices, which can be directly parsed by the optimization solver. Table I gives a simplified example.

TABLE I Structured Instruction Example

Variable	Content
instruction	Generate a feasible disassembly sequence for the treadmill.
input	Candidate task relations: [1,4,10,15], [1,5,11,17], [2,6,12,16], [2,7,13,16], [3,8,14,17], ...
output	[3,8,14,17]

Algorithm 1 LLM-Warm-Started IDOA

Input: Population size N_p, product disassembly information, precedence matrix $s_{pj_1j_2}$, conflict matrix $r_{pj_1j_2}$
Output: Best solution z^*

Begin
Construct structured instructions from product disassembly information.
Generate candidate task sequences by the LLM.
Parse LLM responses into numerical task sequences.
Initialize $P \leftarrow [\,]$.
for each LLM-generated sequence s **do**
  Check precedence and conflict constraints.
  Repair or discard infeasible sequences.
  Insert feasible sequences into P.
end for
while $|P| < N_p$ **do**
  Generate and repair a random individual.
  Insert the individual into P.
end while
Evaluate each individual by Eq. (4).
Evolve the population using IDOA operators.
return the best solution z^*
End

B. LLM-Assisted Population Initialization

The LLM receives the structured instruction and returns one or more candidate task sequences. Since the generated response may contain brackets, commas, spaces, or explanatory text, the raw response is parsed into a numerical sequence before being used by IDOA. The parsed sequence is then checked according to the precedence and conflict constraints extracted from the AND/OR graph. Infeasible sequences are repaired by the decoder or discarded.

After validation, the LLM-generated sequences are inserted into the initial population. The remaining individuals are generated randomly to preserve population diversity. In this way, the initial population contains both LLM-guided individuals and randomly generated individuals. The LLM-guided individuals provide promising starting points, while the random individuals maintain exploration ability.

C. Solution Encoding and Fitness Evaluation

A four-segment encoding scheme $z = \{z_1, z_2, z_3, z_4\}$ is used to represent a complete scheduling solution. Segment z_1 represents the disassembly task sequence. Segment z_2 assigns tasks to factories. Segment z_3 maps tasks to workstations. Segment z_4 specifies the assigned disassembler type. This encoding is consistent with meta-heuristic approaches for disassembly sequence planning and line balancing [24, 25, 26, 27].

Each individual is decoded into a complete schedule, including factory assignment, workstation assignment, tool selection, and disassembler allocation. The fitness value is calculated by the profit-oriented objective function in Eq. (4). The proposed LLM-assisted strategy only modifies the initialization stage. Therefore, the comparison between IDOA and LLM+IDOA directly reflects the effect of LLM-generated initial solutions on convergence behavior and solution quality.

TABLE II Case Information

Case	DL	WM	TM	CB	Tasks	Comp.
1	1	1	0	2	57	69
2	1	0	2	1	58	66
3	2	1	1	0	38	45
4	0	2	1	1	63	72
5	1	1	1	1	54	63

V. Experimental Studies

A. Experimental Setup

The experiments are conducted on five EoL product cases composed of desk lamps, washing machines, treadmills, and car batteries. The maximum number of iterations is set to 200, and the population size is set to 300. The number of disassembly factories is set to 2. Each factory contains 3 workstations, and 5 types of disassembly tools are considered. DL, WM, TM, and CB denote desk lamp, washing machine, treadmill, and car battery, respectively.

B. Comparative Results

The proposed LLM+IDOA is compared with standalone IDOA and CPLEX. CPLEX is used as an exact-solver benchmark, while IDOA is used to evaluate the effect of LLM-assisted initialization. The main metrics are best profit, runtime, and convergence iteration.

As shown in Table III, CPLEX, IDOA, and LLM+IDOA obtain the same best profit in all five cases. This result indicates that LLM-generated sequences do not reduce final solution quality after being refined by IDOA. However, LLM+IDOA requires longer runtime than standalone IDOA in these cases because the runtime includes local LLM inference and sequence generation. Therefore, the advantage of LLM assistance is mainly reflected in initialization quality and convergence behavior rather than wall-clock runtime.

C. Convergence Analysis

Fig. 3 compares the convergence curves of different algorithms for Case 5. LLM+IDOA reaches the best objective value within fewer iterations than standalone IDOA. Specifically, LLM+IDOA converges at approximately the 90th iteration, while IDOA requires about 120 iterations. This result shows that the LLM-generated task sequences provide a more favorable starting point for the subsequent meta-heuristic search.

TABLE III Comparison of CPLEX, IDOA, and LLM+IDOA

Case	Best Profit	Runtime (s)			Convergence Iteration	
		CPLEX	IDOA	LLM+IDOA	IDOA	LLM+IDOA
1	1127	4.330	2.216	9.666	–	–
2	810	4.040	3.283	8.921	–	–
3	1048	1.740	1.596	13.570	–	–
4	1691	4.870	2.772	7.331	–	–
5	1143	4.250	2.301	7.260	120	90

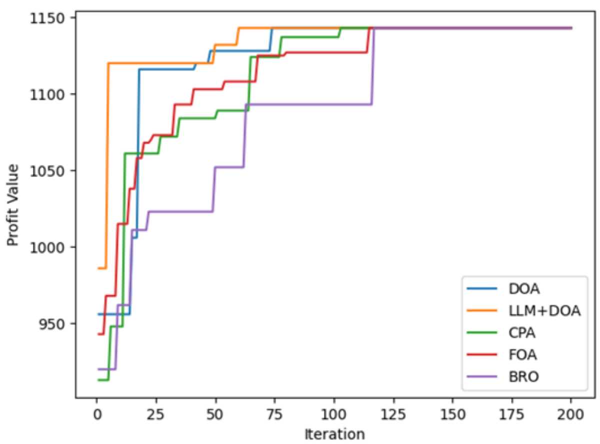

Fig. 3. Convergence curves of different algorithms for Case 5

Although local LLM inference increases total runtime, the convergence curve demonstrates the potential of LLM-assisted initialization. The LLM does not replace IDOA. Instead, it improves the initial population and helps IDOA reach high-quality regions of the search space earlier.

VI. Conclusion

This paper presents an LLM-assisted warm-start strategy for IDOA in multi-factory human-robot collaborative disassembly scheduling. A structured instruction mechanism is designed to guide the LLM in generating candidate task sequences under disassembly constraints. The generated sequences are parsed, validated, and injected into the initial population of IDOA.

Experimental results on five EoL product cases show that LLM+IDOA obtains the same best profit as CPLEX and standalone IDOA. The convergence analysis further indicates that LLM-assisted initialization can reduce the number of iterations required to reach the best objective value. The current implementation introduces additional runtime overhead due to local LLM inference. Nevertheless, the results demonstrate the feasibility of using LLMs as warm-start generators for combinatorial disassembly optimization.

Future work will focus on reducing LLM inference overhead, improving sequence feasibility checking, and extending the proposed method to larger-scale and dynamic disassembly scheduling scenarios.

References

[1] Y. Tang, “Disassembly modeling, planning, and application,” *Journal of Manufacturing Systems*, vol. 21, no. 3, 2002.

[2] M. M. Seamus and M. G. Surendra, *Disassembly Line: Balancing and Modeling*, first edition. ed. New York: McGraw-Hill Education, 2011. [Online]. Available: https://www.accessengineeringlibrary.com/content/book/9780071622875

[3] 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.

[4] 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*, 2023.

[5] 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.

[6] 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.

[7] X. Guo, S. Liu, M. Zhou, and G. Tian, “Disassembly sequence optimization for large-scale products with multiresource constraints using scatter search and petri nets,” *IEEE transactions on cybernetics*, vol. 46, no. 11, pp. 2435–2446, 2015.

[8] ——, “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.

[9] 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.

[10] F. T. Altekin, L. Kandiller, and N. E. Ozdemirel, “Profit-oriented disassembly-line balancing,” *International Journal of Production Research*, vol. 46, no. 10, pp. 2675–2693, 2008.

[11] C. Yang, X. Wang, Y. Lu, H. Liu, Q. V. Le, D. Zhou, and X. Chen, “Large language models as optimizers,” *International Conference on Learning Representations*, 2024.

[12] F. Liu, X. Tong, M. Yuan, X. Lin, F. Luo, Z. Wang, Z. Lu, and Q. Zhang, “Evolution of heuristics: Towards efficient automatic algorithm design using large language model,” in *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[13] H. Ye, J. Wang, Z. Cao *et al.*, “Reevo: Large language models as hyper-heuristics with reflective evolution,” in *Advances in Neural Information Processing Systems*, 2024.

[14] 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.

[15] J. Gu, Z. Guo, J. Wang, L. Qi, S. Qin, and S. Zhang, “Optimization of multi-factory remanufacturing processes with shared transportation resources using the alns algorithm,” *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 1, 2025.

[16] Y. Ji, Z. Zhao, S. Liu, and X. Yong, “Machine learning-based prediction of surplus material in intelligent production processes,” *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 1, 2025.

[17] S. Yan, H. Guo, and S. Liu, “A partition stacking classification framework with oversampling for quality prediction of aluminum alloy ingots,” *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 1, 2025.

[18] T. Lu, L. Qi, W. Luan, K. Liu, X. Guo, and Q. T. A. Talukder, “A novel framework combining vsl and vehicle platooning for freeway

bottleneck," in *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2024, pp. 4061–4066.

[19] Y. Qing and J. Zhao, "Enhanced gold rush optimizer for feature selection application in software fault prediction datasets," *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 2, 2025.

[20] Z. Zhang, S. Dai, C. Li, W. Wang, 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, 2025.

[21] W. Wang, D. Y. Mo, Y. Wang, and M. M. Tseng, "Assessing the cost structure of component reuse in a product family for remanufacturing," *Journal of Intelligent Manufacturing*, vol. 30, no. 2, pp. 575–587, 2019.

[22] A. Koc, I. Sabuncuoglu, and E. Erel, "Two exact formulations for disassembly line balancing problems with task precedence diagram construction using an and/or graph," *IIE Transactions*, vol. 41, no. 10, pp. 866–881, 2009.

[23] G. Tian, M. Zhou, and J. Chu, "A chance constrained programming approach to determine the optimal disassembly sequence," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 4, pp. 1004–1013, 2013.

[24] S. M. McGovern and S. M. Gupta, "A balancing method and genetic algorithm for disassembly line balancing," *European Journal of Operational Research*, vol. 179, no. 3, 2005.

[25] C. B. Kalayci and S. M. Gupta, "A particle swarm optimization algorithm with neighborhood-based mutation for sequence-dependent disassembly line balancing problem," *International Journal of Advanced Manufacturing Technology*, vol. 69, no. 1-4, 2013.

[26] 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.

[27] 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.