World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:16, No:8, 2022

Multi-Robotic Partial Disassembly Line Balancing with Robotic Efficiency Difference via HNSGA-II

Tao Yin, Zeqiang Zhang, Wei Liang, Yanqing Zeng, Yu Zhang

Abstract**—**To accelerate the remanufacturing process of electronic waste products, this study designs a partial disassembly line with the multi-robotic station to effectively dispose of excessive wastes. The multi-robotic partial disassembly line is a technical upgrade to the existing manual disassembly line. Balancing optimization can make the disassembly line smoother and more efficient. For partial disassembly line balancing with the multi-robotic station (PDLBMRS), a mixed-integer programming model (MIPM) considering the robotic efficiency differences is established to minimize cycle time, energy consumption and hazard index and to calculate their optimal global values. Besides, an enhanced NSGA-II algorithm (HNSGA-II) is proposed to optimize PDLBMRS efficiently. Finally, MIPM and HNSGA-II are applied to an actual mixed disassembly case of two types of computers, the comparison of the results solved by GUROBI and HNSGA-II verifies the correctness of the model and excellent performance of the algorithm, and the obtained Pareto solution set provides multiple options for decision-makers.

Keywords**—**Waste disposal, disassembly line balancing, multi-robot station, robotic efficiency difference, HNSGA-II

I. Introduction

With the deterioration of environment and the shortage of natural resources, recycling waste products has become an important means to obtain raw materials for new products. Usually, the waste electronic products contain not only valuable parts that can be reused but also substances harmful to the environment. For example, CPU, memory and hard disk in computers are reusable, while batteries are harmful and need harmless treatment. Therefore, building a complete recycling system can reduce the impact of harmful substances on the environment and the waste of valuable resources. In this system, disassembly is a necessary process to separate parts from the structure of products. Large-scale resource recycling companies generally adopt disassembly line to dispose of wastes. At present, the research on disassembly line includes disassembly line design [1], disassembly line balancing (DLB) [2], and disassembly line sequence planning [3], etc. This study belongs to the field of DLB.

DLB is a classical multi-objective combinatorial optimization problem [4]. Its optimization process is to assign the discrete tasks in the planned sequence to the sequential stations and make the multiple objectives optimal [5]. The existing research on DLB mainly focuses on single manned station mode, that is, only one worker in each station completes all tasks in this station. To improve the internal efficiency of stations, Cevikcan et al. put forward a multi-manned station disassembly line [6]. Considering the line efficiency and the worker safety, Fang et al. proposed a DLB with multi-robotic station (DLBMRS) [7]. Subsequently, Fang, Liu and others made an in-depth study on DLBMRS [8]-[10]. However, all these studies employed complete disassembly mode. Different from the complete disassembly, only the valuable, hazardous and necessary parts are removed, and the remains are directly crushed for raw materials, this is called partial disassembly [11]. Partial disassembly can avoid invalid workload and reduce costs [12]. Thus, this study pays more attention to the partial DLBMRS (PDLBMRS). In reality, the efficiency of robots disassembling different tasks is also different due to the difference of robot models and equipped tools. To get a practical disassembly scheme, this study optimizes the PDLBMRS considering robotic efficiency differences for the first time. The optimization objectives are cycle time of stations, energy consumption of robots, and hazard index of tasks.

In addition, DLB is an NP-hard problem [13]. For the products with small-scale tasks, the common methods include linear programming [14], nonlinear programming [15], and mixed-integer programming [5], etc. While facing the products with large-scale tasks, the mainstream methods are meta-heuristic algorithms, such as hummingbird algorithm [16], firefly algorithm [17], and whale optimization algorithm [18], etc. To effectively solve PDLBMRS, a MIPM which can calculate the single-objective optimal values is established, and an enhanced NSGA-II (HNSGA-II) is proposed. Finally, the MIPM and HNSGA-II are applied to an actual mixed disassembly case of two types of computers, and correctness of the model and excellent performance of the algorithm are verified by comparing their optimization results.

The main contributions of this study are as follows: 1. DLBMRS is expanded from complete disassembly to partial disassembly; 2. Robotic efficiency difference is considered for the first time; 3. MIPM is established to calculate the single-objective global optimal values; 4. HNSGA-II is

This work is supported in part by the National Natural Science Foundation of China under Grant 51205328 and 51675450; in part by the Youth Foundation for Humanities and Social Sciences of Ministry of Education of China under Grant 18YJC630255; in part by the Sichuan Science and Technology Program under Grant 2019YFG0285; and in part by the Special Project of Postgraduate Academic Literacy Improvement Program under Grant 2021KCJS16.

Tao Yin, Zeqiang Zhang*, Wei Liang, Yanqing Zeng, and Yu Zhang are with the Technology and Equipment of Rail Transit Operation and Maintenance Key Laboratory of Sichuan Province, School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China (*corresponding author, e-mail: amniyim@my.swjtu.edu.cn; zhangzq@home.swjtu.edu.cn, liangwei@my.swjtu.edu.cn, zyq2017200281@163.com, chaizhong1997@gmail.com).

proposed to efficiently solve PDLBMRS; 5. MIPM and HNSGA-II are applied to an actual computer disassembly case.

The rest of this paper is organized as follows. Section II describes the PDLBMRS and establishes the MIPM. Section III introduces the HNSGA-II. Section IV employs the model and HNSGA-II to solve an actual mixed disassembly case of two types of computers, and gives the comparison of their optimization schemes. Section V concludes this study and discusses the future research work.

II. Problem Formulation

A. Problem Description

Fig. 1 shows the schematic diagram of PDLBMRS. Two different types of computers enter the disassembly line from the entrance, stations 1-3 are equipped with multiple robots to complete the corresponding tasks. Because of partial disassembly, the parts that do not need to be disassembled will flow out through the outlet of the line and be sent to the crushing workshop for raw materials. After investigation, it is found that most of the existing lines only disassemble a single product, while our designed line can disassemble many types of products.

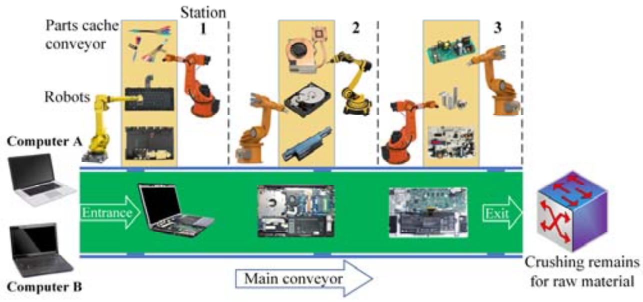


Fig. 1 Schematic diagram of PDLBMRS

B. Notations

a	Product number.
I	Tasks set of products, its cardinality is N_a.
W	Station set, its cardinality is N_w.
R	Robot set, its cardinality is N_r.
i, j	Task number, $i, j \in I$.
w	Station number, $w \in W$.
r	Robot number, $r \in R$.
RL_{max}	Robot limitation in per station.
s_i	Starting time of task i.
t_{ir}	Working time of robot r disassembling task i.
Tc	Cycle time of stations. (s)
OE_r	Operational energy consumption of robot r. (kW·h)
SE_r	Standby energy consumption of robot r.
E_w	Total energy consumption of station w.
B	Large positive number.
$P_a(i)$	Immediate predecessors set of task i in product a.
d_i	Demand attribute. 1, task i is demanded, or 0.
h_i	Hazard attribute. 1, task i is hazardous, or 0.
m_{iwr}	Task assignment variable. 1, task i is assigned to robot r in station w, or 0.
n_{wr}	Robot assignment variable. 1, robot r is assigned to station w, or 0.
k_{ijsw}	Task position variable. 1, tasks i and j are assigned to robot r in station w and j is behind i, or 0.
s_w	Station open variable. 1, station w is opened, or, 0.

C. Optimization Objectives

1) Cycle time: Optimizing cycle time can avoid robots waiting and parts stacking, and make the line smoother and more efficient, so it is regarded as the first objective:

$$\min \ f_1 = Tc \tag{1}$$

2) Energy consumption: As an important factor in the robotic disassembly line, energy consumption objective can be divided into two optimization indexes, namely peak energy consumption of stations and total energy consumption. Their expressions are as follows:

$$\min \ f_2 = \max(E_w) \tag{2}$$

$$\min \ f_3 = \sum_{w \in W} E_w \tag{3}$$

$$E_w = \sum_{r \in R} (\sum_{i \in I} OE_r \cdot m_{iwr} \cdot t_{ir} + SE_r \cdot (Tc - \sum_{i \in I} m_{iwr} \cdot t_{ir})) \ \forall w \in W \tag{4}$$

3) Hazard index: Removing the hazardous parts early can effectively avoid environmental pollution, so another objective is the hazard index:

$$\min \ f_4 = \sum_{i \in I} \sum_{w \in W} \sum_{r \in R} m_{iwr} \cdot (s_i + t_{ir}) \cdot h_i \tag{5}$$

D. Constraints of PDLBMRS

The optimization process of PDLBMRS needs to meet the disassembly mode constraints, precedence relationship constraints, cycle time constraints, task time constraints, task assignment constraints, station configuration constraints, and robot configuration constraints.

1) Disassembly mode constraints: Because of partial disassembly, except for the demanded, hazardous, and necessary tasks, other tasks will not be disassembled. The expressions of partial disassembly are as follows:

$$\sum_{w \in W} \sum_{r \in R} m_{iwr} \le 1 \quad \forall i \in I \tag{6}$$

$$\sum_{w \in W} \sum_{r \in R} m_{iwr} = 1 \quad \forall i \in \{ i | h_i + d_i \ge 1, i \in I \} \tag{7}$$

2) Precedence relationship constraints: Some tasks must be disassembled in order due to the connection relationships or spatial position constraints between parts, this is called precedence relationship. In partial disassembly, if the immediate succeeding task is disassembled, its immediate preceding tasks must be disassembled. While if the immediate preceding task is disassembled, its immediate

succeeding tasks may not be disassembled. This constraint can be expressed as:

$$1 \le \sum_{w\in W}\sum_{r\in R} m_{iwr} + B\cdot(1-\sum_{w\in W}\sum_{r\in R} m_{jwr}) \ \forall j\in I, \forall i \in P_a(j) \quad (8)$$

The precedence relationship constraint is reflected in the timeline that the starting time of the immediate succeeding task must be later than the complete time of its immediate preceding tasks. The expression is shown as:

$$\begin{aligned}&\sum_{w\in W}\sum_{r\in R} m_{jwr}\cdot s_j + B\cdot(1-\sum_{w\in W}\sum_{r\in R} m_{jwr})\\ &\ge \sum_{w\in W}\sum_{r\in R} m_{iwr}\cdot(s_i+t_{ir}) \quad \forall j\in I, \forall i\in P_a(j)\end{aligned} \quad (9)$$

3) Cycle time constraints: First, the working time of each task cannot exceed one cycle time:

$$\sum_{i\in I} m_{iwr}\cdot t_{ir} \le Tc \quad \forall w\in W, \forall r\in R \quad (10)$$

In the process of assigning tasks to stations, the starting time of task *i* disassembled by the current robot in the current station must be large than the sum of the starting time of the current station and the total working time of all tasks assigned to the current robot earlier than task *i*. This constraint can be expressed as:

$$\begin{aligned}&Tc\cdot(\sum_{w\in W}\sum_{r\in R} m_{jwr}\cdot w-1)+\sum_{w\in W}\sum_{r\in R}\sum_{i\in I, i\ne j} k_{ijwr}\cdot t_{ir}\\ &\le \sum_{w\in W}\sum_{r\in R} m_{jwr}\cdot s_j + B\cdot(1-\sum_{w\in W}\sum_{r\in R} m_{jwr}) \ \ \forall j\in I\end{aligned} \quad (11)$$

In addition, the complete time of every task assigned to the current station must be less than the complete time of the current station. The expression is as follows:

$$\sum_{w\in W}\sum_{r\in R} m_{iwr}\cdot(s_i+t_{ir}) \le Tc\cdot(\sum_{w\in W}\sum_{r\in R} m_{iwr}\cdot w) \quad \forall i\in I \quad (12)$$

4) Task time constraints: Taking the starting time of the robot assigned to the first station as the starting point of the timeline, so the starting time of any task should be non-negative, and the constraint expression is as follows:

$$\sum_{w\in W}\sum_{r\in R} m_{iwr}\cdot s_i \ge 0 \quad \forall i\in I \quad (13)$$

For any two tasks assigned to the same robot in the same station, the latter task must wait until the former task is completed. This ensures that a robot can only disassemble one task at once:

$$\begin{aligned}&B\cdot(1-k_{ijwr})+B\cdot(1-\sum_{w\in W}\sum_{r\in R} m_{jwr})+\sum_{w\in W}\sum_{r\in R} m_{jwr}\cdot s_j\\ &\ge \sum_{w\in W}\sum_{r\in R} m_{iwr}\cdot(s_i+t_{ir}) \quad \forall i,j\in I, i\ne j, \forall w\in W, \forall r\in R\end{aligned} \quad (14)$$

5) Task assignment constraint: If two tasks are assigned to the same robot in a station, these two tasks cannot be assigned to other robots or other stations. This can be constrained as:

$$\begin{aligned}&m_{iwr}+m_{jwr} \le 1+(k_{ijwr}+k_{jiwr})\\ &\forall i,j\in I, i<j, \forall w\in W, \forall r\in R\end{aligned} \quad (15)$$

$$\begin{aligned}&0.5\cdot(m_{iwr}+m_{jwr}) \ge k_{ijwr}+k_{jiwr}\\ &\forall i,j\in I, i<j, \forall w\in W, \forall r\in R\end{aligned} \quad (16)$$

6) Station configuration constraints: Although the number of stations is initially given as N_w, it varies during the optimization process. When all tasks are assigned to one station, only the first station needs to be opened. When there are many tasks in the given stations, they need to be all opened. The constraint of the number of opened stations is as follows:

$$1 \le \sum_{w\in W} s_w \le N_w \quad (17)$$

The number of tasks that can be assigned to each station ranges from 0 to N_a. When tasks are assigned to a station, the station is opened; otherwise, the station is closed. This constraint can be expressed as:

$$s_w \le \sum_{i\in I}\sum_{r\in R} m_{iwr} \le s_w\cdot N_a \quad \forall w\in W \quad (18)$$

In addition, according to the layout requirements of the disassembly line, stations should be opened sequentially:

$$s_w \le s_{w-1} \quad \forall w\in W, w\ne 1 \quad (19)$$

7) Robot configuration constraints: In the given N_r robots, some robots can be employed or not:

$$\sum_{w\in W} n_{wr} \le 1 \quad \forall r\in R \quad (20)$$

When a task is assigned to a robot in a station, the robot must be employed in the same station:

$$\sum_{i\in I} m_{iwr} - B\cdot n_{wr} \le 0 \quad \forall w\in W, \forall r\in R \quad (21)$$

There is an upper bound and a lower bound on the number of robots employed in each station. The station without robots will not be opened. The number of robots assigned in an opened station ranges from 1 to RL_{max} due to space limitations. This can be constrained as:

$$0 \le \sum_{r \in R} n_{wr} \le RL_{max} \quad \forall w \in W \tag{22}$$

III. Proposed HNSGA-II

NSGA-II is a classic multi-objective meta-heuristic algorithm with its simple structure and excellent performance [19]. Due to the problem characteristics of PDLBMRS, the crossover and mutation operations of NSGA-II need to be improved. To pursue better solutions, a spur strategy is proposed to enhance NSGA-II. The enhanced algorithm is called HNSGA-II. The structure of HNSGA-II includes encoding and decoding, new solution generation operation (crossover, mutation, and spur strategy), and update population.

A. Encoding and Decoding

In the optimization process, apart from the necessary disassembly task sequence (*DT*), it also needs to construct the station numbering sequence (*SN*) and robot numbering sequence (*RN*) to present the open status of stations and the employed status of robots, respectively. The *DT* can be generated according to the precedence relationships of products. The *SN* size is the number of total tasks N_a, and its elements are the station numbers sorting in ascending order. The *RN* size is the total given robots N_r, its elements are the random arrangement of multiple zeros and station numbers, its index is the robot number, and the element zero indicates that the robot corresponding to the index is not employed. An example of the three sequences meeting the requirements is shown in Fig. 2. The red annotations " *d* " and " *h* " denote the demanded and hazardous attributes of tasks.

Decoding consists of two parts. One is to assign tasks and robots to stations. The assignment result of *DT* and *RN* in Fig. 2 is shown in Fig. 3. Because of partial disassembly, the tasks {5,11,3,9} are not disassembled and station 3 is not opened. The tasks and robots assigned to station 1 are {1,6,2} and {3,7,8}, and those assigned to station 2 are {4,7,8,10} and {2,5,9}.

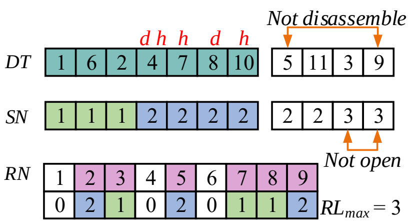


Fig. 2 An example of *DT*, *SN*, and *RN*

Station 1	*Station 2*
$V_{t1} = \{1, 6, 2\}$	$V_{t2} = \{4, 7, 8, 10\}$
$V_{r1} = \{3, 7, 8\}$	$V_{r2} = \{2, 5, 9\}$

Fig. 3 The first part of decoding

Based on the results in Fig. 3, another part of the decoding is to assign the tasks in each station to the robots in the corresponding stations. Taking station 1 as an example to explain the assignment approach: Task 1 will be assigned to the robot which has the shortest disassembly working time in robot set {3,7,8}. If the time disassembling task 1 by robots 3, 7, 8 is the same, the robot with the least operational energy consumption will be selected to disassemble task 1. The assignment of other tasks is the same as that of task 1, and also needs to obey the constraints in the previous section.

B. New Solution Generation Operation

Similar to the original genetic algorithm, the three sequences *DT*, *SN*, *RN* are simultaneously performed the crossover and mutation operations to generate three new sequences. It is worth noting that: a. The new *DT* needs to meet the precedence relationship constraints; b. The new *SN* needs to be sorted in ascending order; c. The new *RN* needs to meet the limitation of RL_{max}. When the newly generated sequences do not meet the relevant constraints, repair method should be used to improve the crossover and mutation operations. In addition, to obtain better solutions, the current non-inferior solution set in each iteration and the four single-objective optimal solutions in this set are selected as parents to be performed the crossover operation, the selection method is called spur strategy. Because the parents are excellent, the new solutions generated by the spur strategy are also considered to be excellent. Thus, the spur strategy is considered as an effective method to enhance the original NSGA-II.

C. Update Population

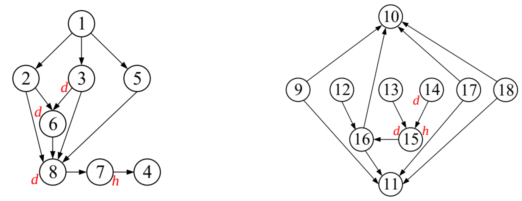
Non-dominated sorting approach and crowding distance of the original NSGA-II are employed to update the population. Besides, an external storage *E* is designed to screen the required number of non-inferior solutions, and the screening method is the crowding distance.

IV. Application to Computer Disassembly Line

The established MIPM and the proposed HNSGA-II are applied to an actual mixed disassembly case of two types of computers. The model is developed by the exact solver GUROBI, and the HNSGA-II is programmed by MATLAB 2014a. Their running environment is Win10 system with an Intel (R) Core (TM) i5-9400 2.9 GHz and 8 GB RAM. By comparing the optimization results of these two methods, correctness of the model and superiority of the algorithm are verified.

A. Data Preparation

The precedence relationships of computer A and computer B come from the literature [20] and are shown in Fig. 4. The number of tasks in computers A and B is 8 and 10, so the total number is 18. Because the efficiency of 16 given robots is different, the time for 16 robots to disassemble the 18 tasks is also different, and the disassembly time is shown in Table I.



(a) Computer A (b) Computer B

Fig. 4 Precedence relationships of two computers

TABLE I
WORKING TIME OF 16 ROBOTS DISASSEMBLING18 TASKS

		Robots															
	t_{ib}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Tasks	1	2	11	14	10	6	11	14	3	14	3	3	3	3	4	10	7
	2	10	6	10	8	9	7	7	8	15	6	2	6	8	15	7	14
	3	15	10	3	15	12	11	4	10	7	13	4	14	13	4	14	11
	4	14	4	6	3	13	14	13	4	9	10	14	11	15	5	5	11
	5	2	8	10	7	6	7	6	10	12	9	6	12	9	11	13	6
	6	11	2	9	10	6	13	6	14	12	6	9	9	13	4	16	3
	7	7	5	5	4	7	4	5	5	8	4	9	4	10	3	6	6
	8	4	6	5	15	11	9	3	4	7	6	10	15	13	11	12	5
	9	10	9	13	11	14	11	10	3	6	4	7	14	2	2	11	7
	10	8	5	11	15	13	12	8	11	3	6	8	8	9	16	8	14
	11	9	7	4	11	14	6	13	2	8	4	8	14	6	10	9	4
	12	14	12	12	2	14	6	9	6	4	14	16	7	6	7	15	4
	13	4	8	8	6	4	15	7	16	8	16	3	7	16	12	15	9
	14	8	3	8	4	14	11	3	13	15	5	8	13	5	15	12	10
	15	3	7	12	11	15	3	5	5	9	12	13	3	9	7	7	4
	16	7	9	8	3	13	11	10	12	15	3	10	15	5	12	3	11
	17	12	14	2	11	14	5	8	4	3	11	6	2	10	10	10	14
	18	3	6	3	9	3	13	15	11	16	11	15	6	16	13	10	13

Energy consumption of robots is divided into operational energy consumption (OE_r) and standby energy consumption (SE_r) per unit time. The energy consumption data of 16 robots are shown in Table II. In addition, the number of the given stations N_w is 4, and the robot limitation in per station RL_{max} is 3.

TABLE II
ENERGY CONSUMPTION DATA OF 16 ROBOTS

Robot	1	2	3	4	5	6	7	8
OE	6.32	6.57	5.92	7.24	6.63	6.40	9.66	7.00
SE	0.63	0.66	0.59	0.72	0.66	0.64	0.97	0.70
Robot	9	10	11	12	13	14	15	16
OE	6.90	7.96	5.34	6.03	8.62	7.88	6.00	9.22
SE	0.69	0.80	0.53	0.60	0.86	0.79	0.60	0.92

B. Optimization Results and Analysis

1) Optimization results: The algorithm parameters are set: population size M = 300, total number of iterations N = 500, and external storage size N_E = 10. After running the HNSGA-II 10 times, one of the obtained optimal disassembly scheme sets is listed in Table III.

TABLE III
AN OPTIMAL DISASSEMBLY SCHEME SET OBTAINED BY HNSGA-II

		Disassembly schemes																	
No.		Must disassemble										Not disassemble							
1(b)	*DT*	13	1	2	14	3	6	5	8	15	7	12	4	9	17	16	18	10	11
	SN	1	1	1	2	2	2	3	3	4	4	4	4	4	4	4	4	4	4
	RN	3	2	2	3	0	4	3	0	0	4	1	1	2	4	0	1		
2(c)	*DT*	1	13	2	5	3	14	6	15	8	7	9	12	18	16	17	11	10	4
	SN	1	1	1	1	2	2	2	3	3	4	4	4	4	4	4	4	4	4
	RN	1	2	2	0	3	2	0	3	1	4	1	3	0	4	4	0		
3(a)	*DT*	1	13	5	14	2	3	15	6	8	7	4	12	16	9	17	18	10	11
	SN	1	1	1	1	1	2	2	2	3	4	4	4	4	4	4	4	4	4
	RN	1	2	2	1	3	0	0	3	0	0	1	2	4	4	3	4		
4	*DT*	14	1	13	3	5	2	15	6	8	7	4	12	9	16	18	17	10	11
	SN	1	1	1	1	1	1	1	2	2	3	3	3	3	3	3	4	4	4
	RN	1	2	0	4	0	0	1	2	2	3	1	3	4	0	3	4		
5	*DT*	14	1	13	3	5	2	15	6	8	7	4	18	12	9	16	17	11	10
	SN	1	1	1	1	1	1	1	2	2	3	3	3	3	3	3	3	3	4
	RN	1	2	3	0	0	0	1	2	0	4	1	3	4	3	2	4		
6	*DT*	13	14	1	15	2	3	5	6	8	7	18	9	12	16	4	17	10	11
	SN	1	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	4
	RN	1	1	4	0	4	3	2	0	3	0	1	0	4	2	3	2		
7	*DT*	14	13	1	15	2	3	6	5	8	7	17	12	16	4	9	18	10	11
	SN	1	1	1	1	1	1	2	2	2	2	2	2	2	3	4	4	4	4
	RN	2	2	0	1	4	3	0	3	0	3	1	1	4	2	4	0		
8	*DT*	13	14	1	2	15	3	6	5	8	7	18	9	12	17	16	4	11	10
	SN	1	1	1	1	1	1	1	1	2	2	3	3	3	3	3	3	4	4
	RN	1	1	3	0	4	3	4	2	0	4	1	2	2	0	0	3		
9	*DT*	14	13	1	2	15	3	5	6	8	7	18	9	4	12	17	16	10	11
	SN	1	1	1	1	1	1	1	1	2	2	2	2	2	2	3	3	4	4
	RN	1	1	3	3	0	4	0	2	2	4	1	0	3	2	0	4		
10(d)	*DT*	13	14	1	15	2	3	5	6	8	7	18	9	4	12	17	16	10	11
	SN	1	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3	4	4
	RN	1	1	3	0	4	3	2	0	3	0	1	0	4	2	4	2		

The Pareto solution set corresponding to the 10 optimal schemes in Table III and the four single-objective global optimal values solved by GUROBI are shown in Table IV. The bold numbers in Table IV indicate the single-objective global optimal values.

TABLE IV
THE RESULTS BY GUROBI AND HNSGA-II

Method	*No.*	f_1	f_2	f_3	f_4	Time/s
GUROBI	1(a)	**5**	-	-	-	197.65
	2(b)	-	**51.79**	-	-	262.90
	3(c)	-	-	**177.61**	-	123.30
	4(d)	-	-	-	**23**	485.77
HNSGA-II	1(b)	**5**	**51.79**	188.93	36	115.03
	2(c)	**5**	52.61	**177.61**	31	
	3(a)	**5**	82.29	189.56	26	
	4	7	139.62	212.08	25	
	5	7	139.62	212.96	24	
	6	9	117.23	213.59	24	
	7	9	118.6	204.55	25	
	8	11	132.69	193.91	25	
	9	11	132.69	195.55	24	
	10(d)	11	132.69	199.39	**23**	

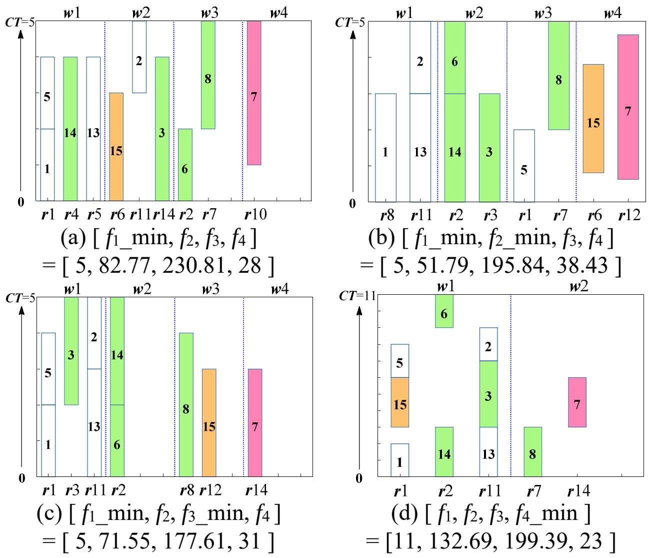
It is observed from Table IV that the HNSGA-II can obtain a Pareto solution set in a single calculation, and the set includes four single-objective global optimal values, which proves the correctness of the model and HNSGA-II. Besides, it is found that the time of obtaining a Pareto solution set is 115.03 s, which is less than the time of obtaining four single-objective global optimal values by GUROBI. This indicates that the efficiency of HNSGA-II for PDLBMRS is higher than GUROBI, and further shows the superiority of HNSGA-II.

2) Disassembly scheme analysis: Figs. 5 and 6 show the Gantt diagrams of disassembly schemes corresponding to the four single-objective global optimal values (a-d) obtained by GUROBI and HNSGA-II. Thereinto, w represents the opened station, r represents the employed robots, the green, pink, yellow and transparent rectangles represent the demanded, hazardous, both demanded and hazardous, and normal tasks. From Figs. 5 and 6, it can be found that except for the same scheme of optimal hazard index f_4_min, schemes of the other three optimal objectives obtained by HNSGA-II are all superior to those of GUROBI. This shows the superiority of HNSGA-II from the quality of solutions. The reason for the results is that GUROBI can only optimize a single objective one by one, while HNSGA-II can optimize four objectives simultaneously.

It is worth mentioning that the Pareto set containing several non-inferior solutions provides rich options for decision-makers. When the decision-makers need to select one optimal scheme, weighting the Pareto solution set is an effective screening method.

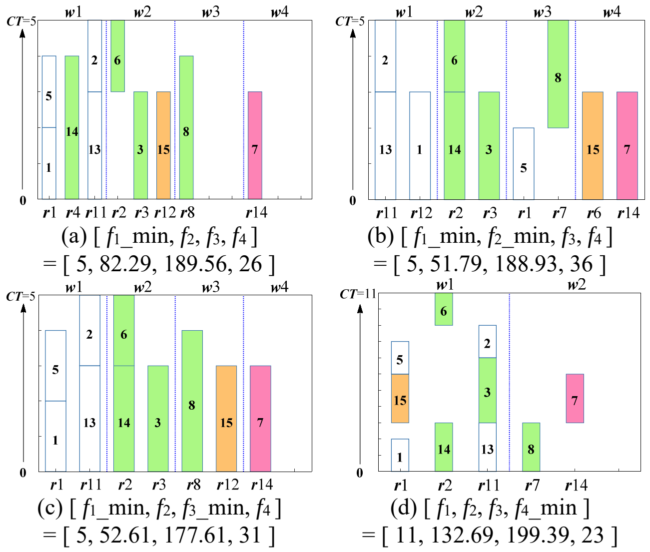
V. Conclusions and Future Research Work

The correctness of PDLBMRS model and the superiority of HNSGA-II are verified by optimizing the mixed disassembly case of two types of computer, which further indicates the established model can perfectly express the objectives and constraints of PDLBMRS.



(a) [f_1_min, f_2, f_3, f_4] = [5, 82.77, 230.81, 28]
(b) [f_1_min, f_2_min, f_3, f_4] = [5, 51.79, 195.84, 38.43]
(c) [f_1_min, f_2, f_3_min, f_4] = [5, 71.55, 177.61, 31]
(d) [f_1, f_2, f_3, f_4_min] = [11, 132.69, 199.39, 23]

Fig. 5 Four single-objective global optimal schemes by GUROBI



(a) [f_1_min, f_2, f_3, f_4] = [5, 82.29, 189.56, 26]
(b) [f_1_min, f_2_min, f_3, f_4] = [5, 51.79, 188.93, 36]
(c) [f_1_min, f_2, f_3_min, f_4] = [5, 52.61, 177.61, 31]
(d) [f_1, f_2, f_3, f_4_min] = [11, 132.69, 199.39, 23]

Fig. 6 Four single-objective global optimal schemes by HNSGA-II

Future research work: 1. PDLBMRS will be expanded from the straight line to the U-shape; 2. Considering the end-of-life state of products will make the PDLBMRS more practical; 3. Each station with multiple robots and multiple workers will be a new disassembly mode in DLB.

References

[1] G. Lowe and R. Bogue, “Design for disassembly: A critical twenty first century discipline,” *Assem. Autom.*, vol. 27, no. 4, pp. 285–289, Oct.2007.
[2] K. Wang, X. Li, L. Gao, and P. Li, “Modeling and balancing for disassembly lines considering workers with different efficiencies,” *IEEE Trans. Cybern.*, to be published.
[3] 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 Trans. Cybern.*, vol. 46, no. 11, pp. 2435–2446, Nov. 2016.
[4] L. Zhu, Z. Zhang, Y. Wang, and N. Cai, “On the end-of-life state oriented multi-objective disassembly line balancing problem,” *J. Intell. Manuf.*, vol. 31, no. 6, pp. 1403–1428, Dec. 2019.
[5] 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,” *Robot. Comput. Integr. Manuf.*, to be published.
[6] E. Cevikcan, D. Aslan, and F. B. Yeni, “Disassembly line design with multi-manned workstations: a novel heuristic optimisation approach,” *Int. J. Prod. Res.*, vol. 58, no. 3, pp. 649–670, Mar. 2019.
[7] Y. Fang, H. Wei, Q. Liu, Y. Li, Z. Zhou, and D. T. Pham, “Minimizing energy consumption and line length of mixed-model multi-robotic disassembly line systems using multi-objective evolutionary optimization,” in *Proc. 14th Int. Manuf. Sci. Eng. Conf. MSEC 2019*, Erie, PA, USA, 2019, pp. 1–10.
[8] Y. Fang, Q. Liu, M. Li, Y. Laili, and D. T. Pham, “Evolutionary many-objective optimization for mixed-model disassembly line balancing with multi-robotic workstations,” *Eur. J. Oper. Res.*, vol. 276, no. 1, pp. 160–174, Dec. 2018.
[9] Q. Liu, Y. Li, Y. Fang, Y. Laili, P. Lou, and D. T. Pham, “Many-objective best-order-sort genetic algorithm for mixed-model multi-robotic disassembly line balancing,” in *Proc. 11th CIRP Conf. IPS*, Zhuhai, China, 2019, pp. 14–21.
[10] H. Ming, Q. Liu, and D. T. Pham, “Multi-robotic disassembly line balancing with uncertain processing time,” in *Proc. 11th CIRP Conf. IPS*, Zhuhai, China, 2019, pp. 71–76.
[11] L. Zhu, Z. Zhang, and C. Guan, “Multi-objective partial parallel disassembly line balancing problem using hybrid group neighbourhood search algorithm,” *J. Manuf. Syst.*, vol. 56, no. 6, pp. 252–269, Jul. 2020.
[12] K. Wang, X. Li, and L. Gao, “Modeling and optimization of

multi-objective partial disassembly line balancing problem considering hazard and profit," *J. Clean. Prod.*, vol. 211, pp. 115–133, Feb. 2019.

[13] S. M. McGovern and S. M. Gupta, "A balancing method and genetic algorithm for disassembly line balancing," *Eur. J. Oper. Res.*, vol. 179, no. 3, pp. 692–708, Jun. 2007.

[14] K. Igarashi, T. Yamada, S. M. Gupta, M. Inoue, and N. Itsubo, "Disassembly system modeling and design with parts selection for cost, recycling and CO2 saving rates using multi criteria optimization," *J. Manuf. Syst.*, vol. 38, pp. 151–164, Jan. 2016.

[15] M. L. Bentaha, O. Battaïa, A. Dolgui, and S. J. Hu, "Second order conic approximation for disassembly line design with joint probabilistic constraints," *Eur. J. Oper. Res.*, vol. 247, no. 3, pp. 957–967, Dec. 2015.

[16] T. Yin, Z. Zhang, and J. Jiang, "A Pareto-discrete hummingbird algorithm for partial sequence-dependent disassembly line balancing problem considering tool requirements," *J. Manuf. Syst.*, vol. 60, pp. 406–428, Jul. 2021.

[17] L. Zhu, Z. Zhang, and Y. Wang, "A Pareto firefly algorithm for multi-objective disassembly line balancing problems with hazard evaluation," *Int. J. Prod. Res.*, vol. 56, no. 24, pp. 7354–7374, Dec. 2018.

[18] Y. Zhang, Z. Zhang, C. Guan, and P. Xu, "Improved whale optimisation algorithm for two-sided disassembly line balancing problems considering part characteristic indexes," *Int. J. Prod. Res.*, to be published.

[19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[20] S. M. McGovern and S. M. Gupta, "Ant colony optimization for disassembly sequencing with multiple objectives," *Int. J. Adv. Manuf. Technol.*, vol. 30, no. 5–6, pp. 481–496, Sep. 2006.