Applied Soft Computing 107 (2021) 107404

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A genetic simulated annealing algorithm for parallel partial disassembly line balancing problem

Kaipu Wang [a], Xinyu Li [a], Liang Gao [a,*], Peigen Li [a], Surendra M. Gupta [b]

[a] State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[b] Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA 02115, USA

ARTICLE INFO

Article history:
Received 24 August 2019
Received in revised form 5 January 2021
Accepted 5 April 2021
Available online 20 April 2021

Keywords:
Parallel disassembly line
Partial disassembly
Stochastic disassembly time
Genetic algorithm
Simulated annealing algorithm

ABSTRACT

The timely recovery and disassembly of waste electrical and electronic equipment (WEEE) can not only obtain a higher economic benefit but also can reduce the impact of hazardous substances on the environment. The parallel disassembly line can disassemble different kinds of WEEE synchronously and improve disassembly efficiency. Therefore, a parallel partial disassembly line balancing model with stochastic disassembly time is established in this paper. The evaluation indexes of the disassembly line include the number of workstations, workload smoothness, and disassembly profits. A new genetic simulated annealing algorithm is proposed to optimize the model. The encoding and decoding strategies are constructed according to the characteristics of partial disassembly and parallel layout. Two-point mapping crossover and single-point insertion mutation operations are designed to ensure that the disassembly sequence meets the precedence constraints and disassembly constraints. The simulated annealing operation is applied to the results of the genetic operation. The proposed algorithm obtains better solutions than the tabu search algorithm in stochastic parallel assembly line balancing problems, and the proposed algorithm has better performance than the CPLEX solver, genetic algorithm, and simulated annealing in parallel disassembly line balancing problems. Finally, a parallel partial disassembly line for waste televisions and refrigerators is constructed, and the performance of the proposed multi-objective algorithm is superior to those of five classical multi-objective algorithms. The results show that the proposed model has a better practical application ability and that the proposed algorithm can improve the performance of disassembly lines.

1. Introduction

Technological innovation has accelerated the speed of product renewal, resulting in a large quantity of waste electrical and electronic equipment (WEEE). There are many types of WEEE, and the most common include used televisions, refrigerators, washing machines, and computers [1]. WEEE occupies various resources and has the risk of environmental pollution. The recovery and disassembly of WEEE can not only create economic benefits but can also be a form of green manufacturing [2]. Disassembly lines are the best way to handle large-scale WEEE [3]. To improve the efficiency of disassembly lines, it is necessary to optimize the disassembly line balancing (DLB) problem.

A reasonable production line layout can be adopted to improve the disassembly efficiency of WEEE. Parallel production lines allow workers to process the same or different products on two adjacent lines [4], which can reduce the idle time of workers and speed up the offline time of multiple products. Therefore, parallel disassembly lines have become one of the better choices for disassembly companies [5]. The quality of used products, such as the rust deformation of parts, is often uncertain, which ultimately leads to uncertainty of disassembly time [6]. All parts on the assembly line must be assembled as required [7], but the disassembly process is different from the assembly process, and the purpose of disassembly is to obtain valuable parts and materials to increase economic efficiency. Thus, a partial disassembly method is often used on disassembly lines [8], and the parts that have not been disassembled are directly crushed and sorted. Harmful parts must be disassembled to reduce their environmental impacts [9]. Therefore, it is essential to construct a new disassembly line balancing model from the layout form and disassembly form and design an efficient optimization method based on the problem characteristics to obtain high-quality disassembly schemes.

There are two main contributions of this paper. First, a parallel partial disassembly line balancing (PPDLB) model is constructed.

* Corresponding author.
E-mail addresses: wangkaipu@hust.edu.cn (K. Wang), lixinyu@mail.hust.edu.cn (X. Li), gaoliang@mail.hust.edu.cn (L. Gao), pgli@mail.hust.edu.cn (P. Li), s.gupta@northeastern.edu (S.M. Gupta).

https://doi.org/10.1016/j.asoc.2021.107404
1568-4946/© 2021 Elsevier B.V. All rights reserved.

Stochastic numbers are used to describe the uncertainty of the disassembly time. The disassembly task variable also has uncertainty. The model uses the number of stations, workload balance, and disassembly profits as the evaluation indexes of the disassembly line. Second, a new genetic simulated annealing algorithm is proposed. The algorithm combines the parallelism of the disassembly line and the uncertainty of the disassembly task to construct the encoding and decoding strategies and incorporates the genetic operation and simulated annealing operation to enhance the global search and local search of the algorithm while satisfying multiple disassembly constraints.

The remainder of this paper is organized as follows: Section 2 is the literature review, Section 3 describes the problem characteristics and mathematical model of the stochastic parallel partial disassembly line, Section 4 designs a genetic simulated annealing algorithm, Section 5 presents the computational experiments and comparisons, and Section 6 gives a summary and future research directions.

2. Literature review

Research on DLB has resulted in a series of research results [10], mainly focusing on the production line layout, disassembly type, disassembly time, optimization objective, optimization method, and multi-objective processing method. A classification of the research literature is shown in Table 1.

The layouts of disassembly lines include linear, U-shaped [11], two-sided [12], and parallel [5] lines. A straight production line has a simple structure and makes it easy to assign tasks. Compared with a straight line, a U-shaped line and a parallel line can further improve disassembly efficiency [13]. A two-sided line is mainly suitable for the disassembly operations of large equipment such as automobiles and engineering machinery [12]. Workers in parallel disassembly lines can simultaneously disassemble multiple products. There are many types of waste electrical and electronic equipment in a disassembly enterprise, so the parallel disassembly line is more practical.

The main types of disassembly include partial disassembly and complete disassembly. If a product is completely disassembled, its essence is the same as its assembly. However, disassembly is not the reverse of assembly because not all parts are disassembled on an actual disassembly line [14]. Only the high-value and hazardous parts are removed on the actual disassembly line, the purpose of which is to increase the disassembly profits and reduce the environmental impacts [9]. Parts with low value are not disassembled but are directly crushed and sorted. Therefore, partial disassembly is more in line with the needs of disassembly enterprises.

The quality of waste products is difficult to determine, so the disassembly process is uncertain [15,16]. In addition to the uncertainty of the choice of disassembled parts, another form of uncertainty in the disassembly process is the disassembly time, which mainly includes stochastic time [17–19] and fuzzy time [20,21]. Compared with the deterministic disassembly time, the uncertain disassembly time can more accurately describe the actual disassembly.

The optimization objectives of the DLB mainly include the number of workstations, the smoothness index, the disassembly demands and hazards, and the number of changes in disassembly direction [22]. The first two objectives are the main evaluation indicators of disassembly line performance [23]. Disassembly enterprises pay more attention to economic benefits, so disassembly profits have become one of the main optimization objectives [24–26]. As energy savings and emission reductions are gradually being concerned by governments, environmentalists, customers, and manufacturing enterprises, disassembly energy consumption has become an optimization objective of disassembly lines [1,27]. The analysis shows that DLB has the characteristics of being multi-objective, multi-constraint, and uncertain.

The optimization methods of DLB mainly include exact methods, heuristic methods, and meta-heuristic methods [28]. Exact methods can theoretically obtain the optimal solution. Due to the limitation of the NP-complete characteristic of the problem [29], exact methods can only be applied to small-scale problems and not to large-scale problems. When using an exact method, first, the mixed-integer programming model is usually established; and then solved using the specific software [30], such as GAMS and CPLEX solver. Furthermore, heuristic methods cannot determine whether a result is an optimal solution, but the methods are simple [31]. Common heuristic methods include greedy algorithms, hill climbing, 2 optimal [29], and hybrid heuristics [32]. For example, Avikal et al. [33,34] used the fuzzy AHP and PROMETHEE, Kano model, AHP, and M-TOPSIS, which can quickly obtain better solutions, to optimize multi-objective DLB.

Although meta-heuristic methods are more complex than heuristic methods, they can obtain near-optimal solutions in a reasonable time, and they are mainly applied to large-scale problems. Common meta-heuristic methods include the genetic algorithm [35], ant colony optimization [23], particle swarm optimization [36], artificial bee colony [37], tabu search [38], and variable neighborhood search [39]. Meta-heuristic methods are still the better choice for the disassembly of large-scale products. However, traditional algorithms ignore the problem constraints and easily produce infeasible solutions. Furthermore, the performance of the basic algorithms, such as tabu search [40,41], can be further improved. To improve the performances of these methods, a hybrid meta-heuristic algorithm can be designed according to the problem characteristics.

The DLB is a multi-objective optimization problem. The methods for handling multiple objectives in the existing literature include weighting [42], lexicographic order [20], and Pareto dominance [43]. The weighting method is relatively simple, but it is difficult to set the weights reasonably. The lexicographic order optimizes the objectives in order of importance and only optimizes the most important objectives while ignoring others [35–39]. The multi-objective processing method based on Pareto dominance does not set the optimization order of objectives and obtains multiple solutions at a time, which can balance each objective [26,43–46]. Because each objective in the disassembly line has its focus, the multi-objective method based on Pareto dominance is more applicable.

In summary, there are two tasks for the DLB of waste products: the first is to establish a comprehensive evaluation model of the disassembly line by adopting a reasonable layout and disassembly type, and the second is to design an efficient optimization method to obtain high-quality disassembly schemes.

3. Stochastic parallel partial disassembly line balancing

3.1. Problem description

The parallel production line allows workers to operate on two adjacent lines. As shown in Fig. 1, workers 1, 3, and 5 all operate simultaneously between two adjacent disassembly lines. Parallel lines can shorten product completion time, improve worker disassembly efficiency, and reduce idle time [52]. The purpose of disassembly is to obtain valuable parts, so there is no need to completely disassemble waste products, that is, partial disassembly is adopted. The tasks in the slashed frame in Fig. 1 are not disassembled but are crushed and sorted directly.

Table 1
Literature analysis of disassembly line balancing problem. Gungor and Gupta [47]; Gungor and Gupta [3]; McGovern and Gupta [23]; McGovern and Gupta [29]; McGovern and Gupta [22]; Agrawal and Tiwari [11]; Altekin, Kandiller and Ozdemirel [8]; Ding, Feng, Tan and Gao [43]; Aydemir-Karadag and Turkbey [42]; Kalayci and Gupta [36]; Kalayci and Gupta [37]; Avikal, Jain and Mishra [33]; Avikal, Mishra and Jain [34]; Bentaha, Battaia and Dolgui [15]; Kalayci and Gupta [38]; Tuncel, Zeid and Kamarthi [16]; Bentaha, Battaia and Dolgui [30]; Bentaha, Battaia, Dolgui and Hu [6]; Hezer and Kara [5]; Kalayci, Hancilar, Gungor and Gupta [20]; Kalayci, Polat and Gupta [39]; Riggs, Battaia and Hu [32]; Altekin, Bayindir and Gumuskaya [17]; Kalayci, Polat and Gupta [35]; Kalaycilar, Azizoglu and Yeralan [48]; Mete, Cil, Agpak, Ozceylan and Dolgui [49]; Altekin [18]; Ren, Yu, Zhang, Tian, Meng and Zhou [25]; Zhang, Wang, Zhu and Wang [21]; Bentaha, Dolgui, Battaia, Riggs and Hu [14]; Kazancoglu and Ozturkoglu [2]; Liu, Zhou, Duc Truong, Xu, Yan, Liu, Ji and Liu [45]; Pistolesi, Lazzerini, Mura and Dini [26]; Ren, Zhang, Zhao, Tian, Lin, Meng and Li [31]; Zheng, He, Chu and Liu [19]; Zhu, Zhang and Wang [46]; Fang, Liu, Li, Laili and Duc Truong [44]; Li, Chen, Zhu, Yang and Chu [50]; Mete, Cil, Celik and Ozceylan [51]; Wang, Li and Gao [9]; Wang, Li and Gao [12]; Wang, Li, Gao and Garg [1].

No.	Year	DLT				DT		OT			OF										OM			MOPM		
		SDL	UL	TL	PL	CD	PD	DDT	ST	FT	CT	DEF	NS	SI	DH	DEM	DIR	DC	DP	DE	EM	HM	MHM	WM	LM	PM
[47]	2001	✓				✓		✓					✓									✓				
[3]	2002	✓				✓		✓					✓									✓				
[23]	2006	✓				✓		✓					✓	✓	✓	✓	✓						ACO		✓	
[29]	2007	✓				✓		✓				✓	✓	✓	✓	✓	✓					✓	ACO + GA		✓	
[22]	2007	✓				✓		✓					✓	✓	✓	✓	✓					✓	GA		✓	
[11]	2008		✓			✓			✓			✓		✓									ACO		✓	
[8]	2008	✓					✓	✓											✓		✓					
[43]	2010	✓				✓		✓					✓	✓		✓							ACO			✓
[42]	2013	✓				✓			✓					✓				✓					GA	✓		
[36]	2013	✓				✓		✓					✓	✓		✓							PSO		✓	
[37]	2013	✓				✓		✓					✓	✓	✓	✓							ABC		✓	
[33]	2014	✓				✓		✓			✓			✓		✓						✓				
[34]	2014	✓				✓		✓			✓			✓								✓				
[15]	2014	✓				✓			✓									✓			✓					
[38]	2014	✓				✓		✓					✓	✓	✓	✓							TS		✓	
[16]	2014	✓				✓		✓						✓	✓	✓							RL		✓	
[30]	2015	✓				✓			✓									✓			✓					
[6]	2015	✓					✓		✓									✓			✓					
[5]	2015				✓	✓		✓					✓									✓				
[20]	2015	✓				✓				✓			✓	✓	✓	✓							ABC		✓	
[39]	2015	✓				✓		✓					✓	✓	✓	✓							VNS		✓	
[32]	2015	✓				✓			✓		✓											✓				
[17]	2016	✓				✓			✓										✓			✓				
[35]	2016	✓				✓		✓					✓	✓	✓	✓							GA		✓	
[48]	2016	✓				✓		✓			✓										✓					
[49]	2016	✓				✓		✓					✓										BS			
[18]	2017	✓				✓			✓				✓								✓					
[25]	2017	✓					✓	✓											✓				GS			
[21]	2017	✓				✓				✓			✓	✓			✓	✓					AFS			✓
[14]	2018	✓					✓		✓										✓		✓					
[2]	2018	✓				✓		✓			✓									✓		✓				
[45]	2018	✓				✓		✓					✓	✓	✓								ABC			✓
[26]	2018	✓					✓	✓					✓						✓				GA			✓
[31]	2018	✓				✓		✓						✓								✓				
[19]	2018	✓					✓		✓									✓			✓					
[46]	2018	✓				✓		✓					✓	✓	✓								FA			✓
[44]	2019	✓				✓		✓			✓		✓	✓									BEA			✓
[50]	2019	✓				✓		✓			✓										✓					
[51]	2019	✓				✓		✓										✓			✓					
[9]	2019	✓					✓	✓					✓	✓					✓				GSA			✓
[12]	2019			✓			✓		✓				✓	✓					✓	✓			FPA			✓
[1]	2019	✓					✓		✓				✓	✓					✓	✓			FPA			✓

DLT: disassembly line type; SDL: straight disassembly line; UL: U-shaped line; TL: two-sided line; PL: parallel line; DT: disassembly type; CD: complete disassembly; PD: partial disassembly; OT: operation time; DDT: determine disassembly time; ST: stochastic time; FT: fuzzy time; OF: objective function; CT: cycle time; DEF: disassembly efficiency; NS: number of stations; SI: smoothness index; DH: disassembly hazard; DEM: disassembly demand; DIR: disassembly direction; DC: disassembly cost; DP: disassembly profit; DE: disassembly energy; OM: optimization method; EM: exact method; HM: heuristic method; MHM: meta-heuristic method; MOPM: multi-objective processing method; WM: weighting method; LM: lexicographic method; PM: Pareto method; ACO: ant colony optimization; GA: genetic algorithm; PSO: particle swarm optimization; ABC: artificial bee colony; TS: tabu search; RL: reinforcement learning; VNS: variable neighborhood search; BS: beam search; GS: gravitational search; AFS: artificial fish swarm; FA: firefly algorithm; BEA: bi-criterion evolutionary algorithm; GSA: genetic simulated annealing; FPA: flower pollination algorithm.

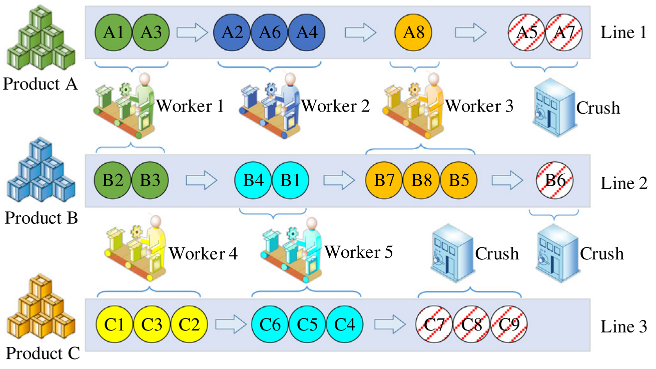


Fig. 1. Parallel disassembly line.

The value of the parts remains the same when the normal disassembly method is adopted, but the parts that are not disassembled can only be recycled after being crushed, and the disassembly value will be reduced. It should be noted that hazardous parts must be disassembled to reduce environmental impacts. Due to the physical structure of the product and disassembly process restrictions, the disassembly sequence of product parts has precedence relationships.

To complete the disassembly tasks within the specified time, the cycle time is set in the disassembly line. The cycle times of different products are different on parallel disassembly lines. Therefore, the cycle time of parallel disassembly lines needs to be redefined to facilitate the assignment of the disassembly tasks. If parallel disassembly lines contain H disassembly lines and the cycle time of line h is set to CT_h, then the common cycle time of all disassembly lines is the minimum common multiple of all cycle times [40], which is recorded as CT. The time coefficient of line h is set to λ_h, and its expression is as follows.

$$\lambda_h = CT/CT_h \tag{1}$$

The object of a disassembly line is large-scale waste products, and the quality and structure of different products are uncertain. This paper uses the stochastic disassembly time to describe this uncertainty. Let the disassembly time of task i on line h be t_{hi}, where $t_{hi} \sim N\ (\mu_{hi},\ \sigma_{hi}^2)$. Then the representation of t_{hi} on the parallel disassembly lines is as follows.

$$t'_{hi} = \lambda_h \cdot t_{hi} \Rightarrow t'_{hi} \sim N(\lambda_h \cdot \mu_{hi},\ \lambda_h^2 \cdot \sigma_{hi}^2) \tag{2}$$

For example, the part information of products A and B, including the mean and standard deviation of disassembly time, and the hazard, are shown in Tables 2 and 3, respectively. If CT_{A} and CT_{B} are set to 50 and 60, respectively, then the least common multiple of the two cycle times is 300, that is, the cycle time of the parallel disassembly lines is 300. According to Eq. (1), λ_{A} and λ_{B} can be calculated as 6 and 5, respectively, and the part information of products A and B on the parallel disassembly lines can be calculated according to Eq. (2), as shown in Table 4.

Disassembly enterprises focus on disassembly efficiency and economic benefits. Therefore, the optimization objectives of parallel partial disassembly line balancing in this paper are to minimize the number of workstations, balance the workload of stations, and maximize the disassembly profits. Parallel disassembly line balancing aims to reasonably select disassembly tasks and to assign them to workstations under the premise of meeting various disassembly constraints while optimizing multiple evaluation indicators of the disassembly line.

To facilitate the mathematical description of the model, this paper sets the following assumptions:

- The cycle time of each line is known.
- The disassembly time of parts is a random number obeying a normal distribution.
- The precedence relationships, value, and harmfulness of the parts are known.
- The quantity of products in each line is sufficient, and the parts are complete and cannot be divided.
- Each worker is versatile and can work between any two adjacent lines.
- Workers' walking time between lines is ignored.

3.2. Mathematical model

The notations in the mathematical model are defined as follows:

Indices:

h, q	disassembly line
i_h, l_h	the task of line h
j, g	workstation

Parameters:

N_h	number of tasks on line h
H	number of disassembly lines
J	number of workstations
μ_{hi}	average disassembly time of task i on line h
σ_{hi}	the standard deviation of the disassembly time of task i on line h
α	the confidence level that the disassembly time does not exceed the cycle time
T_j	disassembly time of workstation j
r_{hi}	revenue of task i on line h
ρ_{hi}	value rate of task i on line h; 1 if the task is disassembled; ρ_0 if the task is crushed, $0 < \rho_0 < 1$
C_f	cost of a single station
C_g	cost of a parallel station
C_s	the unit time cost of a station
ζ, ξ	station coefficient
ψ	large number
CT	cycle time of parallel lines
AT	actual cycle time of parallel lines
NS	number of stations opened
SI	smoothness index of parallel lines
DF	disassembly profit of parallel lines

Decision variables:

$$x_{hi} = \begin{cases} 1, & \text{if task } i \text{ is disassembled on line } h \\ 0, & \text{otherwise} \end{cases}$$

$$y_{hij} = \begin{cases} 1, & \text{if task } i \text{ on line } h \text{ is assigned to workstation } j \\ 0, & \text{otherwise} \end{cases}$$

$$F_j = \begin{cases} 1, & \text{if workstation } j \text{ is utilized on only one line} \\ 0, & \text{otherwise} \end{cases}$$

$$G_j = \begin{cases} 1, & \text{if workstation } j \text{ is utilized on two adjacent lines} \\ 0, & \text{otherwise} \end{cases}$$

$$U_{hj} = \begin{cases} 1, & \text{if workstation } j \text{ is utilized on line } h \\ 0, & \text{otherwise} \end{cases}$$

Auxiliary parameters:

$$D_{hi} = \begin{cases} 1, & \text{if task } i \text{ on line } h \text{ is harmful} \\ 0, & \text{otherwise} \end{cases}$$

$$p_{hil} = \begin{cases} 1, & \text{if task } i \text{ is the immediate predecessor of task} \\ & j \text{ on line } h \\ 0, & \text{otherwise} \end{cases}$$

Table 2
The part information of product A.

Task ID	1	2	3	4	5	6	7	8
Mean (μ)	14	10	12	18	23	16	20	36
Standard deviation (σ)	3.50	2.50	3.00	4.50	5.75	4.00	5.00	9.00
Hazard (D)	0	0	0	0	0	0	0	0

Table 3
The part information of product B.

Task ID	1	2	3	4	5	6	7	8	9	10
Mean (μ)	14	10	12	18	23	16	20	36	14	10
Standard deviation (σ)	3.50	2.50	3.00	4.50	5.75	4.00	5.00	9.00	3.50	2.50
Hazard (D)	0	0	0	0	0	0	1	0	0	0

Table 4
The part information of products A and B on the parallel disassembly line.

Task ID	A1	A2	A3	A4	A5	A6	A7	A8	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10
Mean (μ)	84	60	72	108	138	96	120	216	70	50	60	90	115	80	100	180	70	50
Standard deviation (σ)	21.0	15.0	18.0	27.0	34.5	24.0	30.0	54.0	17.5	12.5	15.0	22.5	28.75	20.0	25.0	45.0	17.5	12.5
Hazard (D)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Referring to the stochastic parallel assembly line balancing model [40] and the partial disassembly line balancing model [12], this paper establishes a stochastic parallel partial disassembly line balancing (SPPDLB) model, as shown below.

The objective functions include the number of workstations, smoothness index, and disassembly profits. Eq. (3) is a comprehensive evaluation of the number of workstations, and its coefficients ζ and ξ are set to 2 and 1. Eq. (4) represents the workload smoothness index, where the station time T_j is calculated using Eq. (5) [40], and AT is the maximum working time of the stations calculated using Eq. (6). Eq. (7) is the disassembly profits, which is equal to the disassembly revenues minus the costs. The disassembly costs include station opening costs and station operating costs.

$$\min f_1 = NS = \xi \cdot \sum_{j=1}^{J} \left(F_j + G_j\right) + \zeta \cdot \sum_{j=1}^{J} \sum_{h=1}^{H} U_{hj} \tag{3}$$

$$\min f_2 = SI = \sqrt{\sum_{j=1}^{J} \left(AT - T_j\right)^2} \tag{4}$$

$$T_j = \sum_{h=1}^{H} \sum_{i=1}^{N_h} x_{hi} y_{hij} \mu_{hi} + \Phi^{-1}(\alpha) \sqrt{\sum_{h=1}^{H} \sum_{i=1}^{N_h} x_{hi} y_{hij} \sigma_{hi}^2} \tag{5}$$

$$AT = \left\lceil \max_{1 \le j \le J} \left\{T_j\right\} \right\rceil \tag{6}$$

$$\max f_3 = DF = \sum_{h=1}^{H} \sum_{i=1}^{N_h} x_{hi} \rho_{hi} r_{hi} - C_f \sum_{j=1}^{J} F_j - C_g \sum_{j=1}^{J} G_j - C_s \cdot AT \sum_{j=1}^{J} \left(F_j + G_j\right) \tag{7}$$

The constraints are as follows: Eq. (8) indicates that there is at least one disassembly task in the opened station, that is, no empty station is allowed. Eq. (9) indicates that hazardous tasks must be disassembled. Tasks without hazards can be disassembled or directly crushed without disassembly, as shown in Eq. (10). The order of the disassembly tasks on the same line needs to meet the precedence constraints, as shown in Eq. (11). A worker can only work between two adjacent lines and cannot work across the lines, as shown in Eqs. (12) and (13) [40,53]. Eq. (14) denotes the relationship between the numbers of workstations. Eq. (17) is the variables in the model, including the disassembly variables, task assignment variables, and station opening variables.

$$\sum_{h=1}^{H} \sum_{i=1}^{N_h} x_{hi} y_{hij} \ge 1, \forall j = 1, \ldots, J \tag{8}$$

$$\sum_{j=1}^{J} x_{hi} y_{hij} = 1, \forall i_h \in \{i_h \,|D_{hi} = 1\} \tag{9}$$

$$\sum_{j=1}^{J} x_{hi} y_{hij} \le 1, \forall i_h \in \{i_h \,|D_{hi} = 0\} \tag{10}$$

$$\sum_{j=1}^{J} j \cdot y_{hij} - \sum_{g=1}^{J} g \cdot y_{hlg} \le 0, \quad \forall (i_h,\ l_h) \in \{(i_h,\ l_h)\,|p_{hil} = 1 \wedge x_{hi} x_{hl} = 1\} \tag{11}$$

$$U_{hj} + U_{(h+q)j} = 1, \forall j = 1, \ldots, J\,;\ h = 1, \ldots, H-2;\ q = 2, \ldots, H-h \tag{12}$$

$$\sum_{i=1}^{N_h} x_{hi} y_{hij} - \psi U_{hj} \le 0, \forall h = 1, \ldots, H\ \ ; j = 1, \ldots, J \tag{13}$$

$$\sum_{h=1}^{H} U_{hj} - 2G_j - F_j = 0, \forall j = 1, \ldots, J \tag{14}$$

$$x_{hi},\ y_{hij},\ F_j,\ G_j,\ U_{hj} \in \{0, 1\}, \forall h,\ i,\ j \tag{15}$$

3.3. A lower bound

Partial disassembly is different from complete disassembly. In addition to hazardous tasks that must be disassembled, other tasks may not be disassembled. Due to the precedence constraints, hazardous tasks can only be disassembled after all of their predecessors have been disassembled. If the immediate task set of task i on line h is S_{hi}, then the set (S^*) of all hazardous tasks and their predecessors on the parallel disassembly lines can be expressed as follows.

$$S^* = \left\{ \bigcup_{h,i_h} S_{hi}, i_h \right\},\ \forall h \in \{1, \ldots, H\};\ i_h \in \left\{ i_h \middle| D_{hi} = 1 \right\} \tag{16}$$

On parallel partial disassembly lines, only when the hazardous tasks and their predecessors are disassembled, the number of opened stations is the lowest. Therefore, the lower bound (LB) of the number of stations on the parallel partial disassembly lines can be calculated as follows.

$$LB = \left\lceil \left(\sum_{h=1}^{H} \sum_{i_h \in S^*} \mu_{hi} + \Phi^{-1}(\alpha) \sqrt{\sum_{h=1}^{H} \sum_{i_h \in S^*} \sigma_{hi}^2} \right) \middle/ CT \right\rceil \tag{17}$$

4. Genetic simulated annealing algorithm

The genetic algorithm has a strong global ability, the simulated annealing algorithm has a strong local ability [41], and a hybrid algorithm can realize the complementary advantages of the two algorithms [9]. The simulated annealing operation is performed on the results of the genetic operation in the hybrid algorithm. In this paper, Pareto dominance is used to evaluate the multi-objective disassembly sequence, and the crowding distance [54] is used to filter non-inferior solutions. The operations of the proposed algorithm consider disassembly constraints, so the operating results are all feasible solutions, avoiding the additional verification and adjustment operations of the traditional algorithm to ensure the feasibility of the results. The genetic simulated annealing algorithm mainly includes initialization, genetic operation, simulated annealing operation, and population update [27]. The number of cooling times in the simulated annealing operation is taken as the termination criterion of the algorithm. This section will describe the operations of the algorithm in detail.

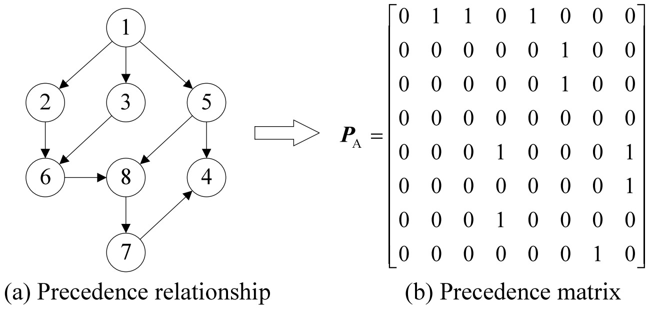
4.1. Encoding strategy

The disassembly sequence for parallel partial disassembly line balancing consists of two parts: the disassembly task sequence and the disassembly variable sequence. The disassembly task sequence sorts all disassembly tasks and requires them to meet the precedence constraints. The disassembly variable sequence is set using binary variables to represent whether the tasks are disassembled.

To ensure the feasibility of the initial solution in the encoding process, a precedence matrix is used to select tasks. The tasks on the parallel lines come from a variety of products, each with different precedence relationships. Therefore, a composite precedence matrix can be constructed to describe the relationships between all product tasks. Let the precedence matrix of line h be $\boldsymbol{P}_h$, and its expression is as follows.

$$\boldsymbol{P}_h = [p_{hil}]_{N_h \times N_h},\ \forall i,\ l = 1, \ldots, N_h;\ h = 1, \ldots, H \tag{18}$$

where p_{hil} is the precedence relationship between parts i and l on line h. If part i is the immediate predecessor of part l, then $p_{hil} = 1$; otherwise, $p_{hil} = 0$. N_h represents the number of tasks on line h, and H indicates the number of lines. The precedence relationships and precedence matrix of products A and B are shown in Figs. 2 and 3, respectively.

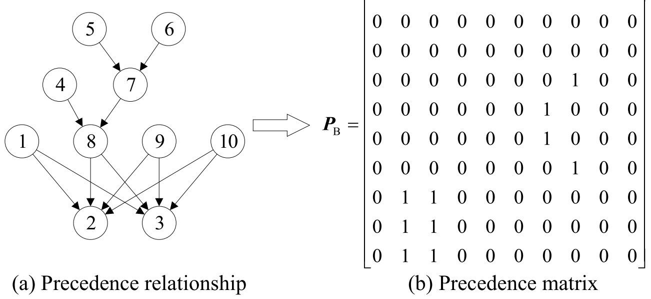


(a) Precedence relationship

$$\boldsymbol{P}_{\mathrm{A}} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

(b) Precedence matrix

Fig. 2. The precedence relationships and precedence matrix of product A.

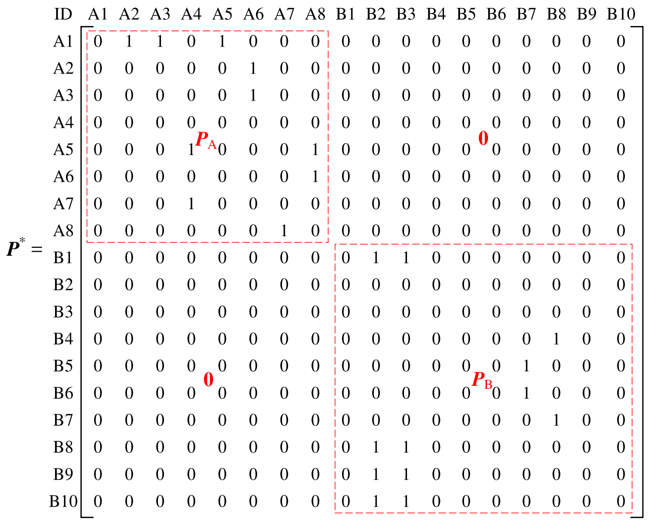


(a) Precedence relationship

$$\boldsymbol{P}_{\mathrm{B}} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) Precedence matrix

Fig. 3. The precedence relationships and precedence matrix of product B.



$$\boldsymbol{P}^* = \begin{array}{c|cccccccccccccccccc|} \text{ID} & \text{A1} & \text{A2} & \text{A3} & \text{A4} & \text{A5} & \text{A6} & \text{A7} & \text{A8} & \text{B1} & \text{B2} & \text{B3} & \text{B4} & \text{B5} & \text{B6} & \text{B7} & \text{B8} & \text{B9} & \text{B10} \\ \text{A1} & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{A2} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{A3} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{A4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{A5} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{A6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{A7} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{A8} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{B1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{B2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{B3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{B4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \text{B5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \text{B6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \text{B7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \text{B8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{B9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{B10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Fig. 4. Composite precedence matrix of products A and B.

The expression of the composite precedence matrix $\boldsymbol{P}^*$ is shown in Eq. (19). The precedence matrices are all on the diagonal of $\boldsymbol{P}^*$, and the other elements in $\boldsymbol{P}^*$ are 0 matrices. The $\boldsymbol{P}^*$ of products A and B is shown in Fig. 4.

$$\boldsymbol{P}^* = \begin{bmatrix} \boldsymbol{P}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{P}_h & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{P}_H \end{bmatrix} \tag{19}$$

The specific steps for the encoding of the disassembly task sequence are as follows:

Step 1: Select the task that has no immediate predecessor or its immediate predecessor has been assigned and add it to the candidate task set. If the task at the first position is selected, the set is {A1, B1, B4, B5, B6, B9, B10}.

Step 2: Randomly select a task from the candidate task set as the disassembly task of the current position, such as task A1.

Table 5
A feasible disassembly sequence of the parallel disassembly line.

Task ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
X_1	B5	B6	B7	A1	B9	A2	B1	A3	B4	A6	B8	A5	B10	A8	B2	A7	B3	A4
X_2	1	1	1	1	1	1	0	1	1	1	0	1	1	0	0	0	0	0

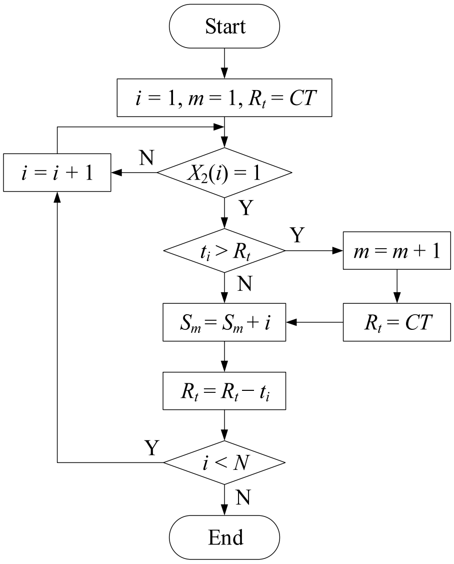


Fig. 5. Decoding operation flow.

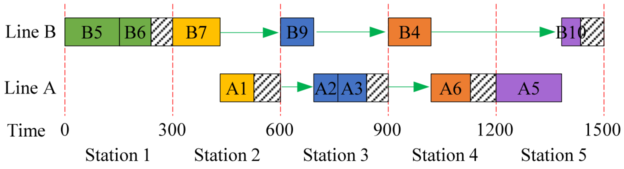


Fig. 6. Gantt chart of the disassembly tasks.

Step 3: Update the precedence matrix. The constraint relationship between the selected task and other tasks is released so that the row and column of task A1 in $\boldsymbol{P}^*$ are 0, that is, $\boldsymbol{P}^*(\mathrm{A1}, \mathrm{A2}) = 0$, $\boldsymbol{P}^*(\mathrm{A1}, \mathrm{A3}) = 0$, and $\boldsymbol{P}^*(\mathrm{A1}, \mathrm{A5}) = 0$.

Step 4: Select the task at the next position and repeat Steps 1 to 3.

Step 5: All tasks are assigned, and the encoding operation of the disassembly task ends.

It is necessary to satisfy constraints (9) to (11) when determining the disassembly task variables. That is, hazardous tasks and their predecessors must be disassembled, and other tasks can be disassembled or not disassembled. The disassembly variable for the disassembled task is 1; otherwise, it is 0.

For example, task B7 is hazardous on the parallel disassembly lines consisting of products A and B, and a feasible disassembly sequence $\boldsymbol{X}$ of the disassembly line is shown in Table 5, where $\boldsymbol{X} = [\boldsymbol{X}_1; \boldsymbol{X}_2]$, $\boldsymbol{X}_1$ represents the disassembly task sequence, and $\boldsymbol{X}_2$ indicates the disassembly task variables.

4.2. Decoding strategy

The decoding operation assigns the tasks that need to be disassembled to each workstation while satisfying the cycle time constraints. Before assigning tasks, it is necessary to determine the cycle time of the parallel lines according to the cycle time of different product lines, and then determine the disassembly time of the tasks on the parallel lines. The decoding operation flow is shown in Fig. 5, where i is the disassembly task, m is the station, CT represents the cycle time, R_t indicates the remaining time of the station, t_i is the disassembly time of task i, S_m represents the disassembly task set in station m, and N indicates the number of tasks.

The confidence level of the station time to satisfy the cycle time constraint is 0.9. Based on the disassembly information in Table 4, the disassembly sequence in Table 5 is decoded according to the flow in Fig. 3, and the decoded task assignment is shown in Fig. 6. The dashed boxes in the figure indicate the idle time of stations. A total of 5 stations are opened, including one unilateral station and four parallel stations. The disassembly tasks and the sequences in each station are the following: $S_1 = \{\mathrm{B5} \rightarrow \mathrm{B6}\}$, $S_2 = \{\mathrm{B7} \rightarrow \mathrm{A1}\}$, $S_3 = \{\mathrm{B9} \rightarrow \mathrm{A2} \rightarrow \mathrm{A3}\}$, $S_4 = \{\mathrm{B4} \rightarrow \mathrm{A6}\}$, and $S_5 = \{\mathrm{A5} \rightarrow \mathrm{B10}\}$. The disassembly time of each station is the following: $T_1 = 239.88$, $T_2 = 225.84$, $T_3 = 239.48$, $T_4 = 228.16$, and $T_5 = 235.03$.

4.3. Fitness evaluation

Section 3.2 shows that the optimization objectives of parallel partial disassembly line balancing include minimizing the number of workstations, minimizing the workload smoothness, and maximizing the disassembly profit. Therefore, the fitness value of the problem can be expressed as follows.

$$\min F = \min\,(NS,\ SI,\ -DF) \tag{20}$$

The number of workstations and disassembly profits are two conflicting sub-objectives. When evaluating the performance of the disassembly sequence, the fitness value cannot be directly compared as a single objective, but the dominance relation is used to describe the advantages and disadvantages of the disassembly sequences.

For any two disassembly sequences S_1 and S_2, if all sub-objectives of S_1 are not worse than those of S_2, and there is at least one sub-objective of S_1 that is better than that of S_2, then S_1 dominates S_2, which is denoted as $S_1 \prec S_2$ [54]. A solution that is not dominated by all other solutions is called a Pareto non-inferior solution. The solution of disassembly line balancing is a set of non-inferior solutions. The objective value of the non-inferior solution set is the Pareto front.

All non-inferior solutions are stored in the external file, which is updated continuously in the algorithm iteration process. In the iterative process of the algorithm, the new solution is compared with the current non-inferior solution, the non-dominated solution is retained, the dominated solution is eliminated, and the external file is updated [54].

4.4. Genetic algorithm

Genetic operations mainly include selection operations, crossover operations, and mutation operations. To ensure that a new solution is feasible, it is necessary to design the operations according to the problem characteristics when performing the crossover and mutation, that is, the new solution is required to satisfy the precedence constraints.

Table 6
Parent disassembly sequence before crossover.

Task ID		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Parent 1	X_1	B5	B6	B7	A1	B9	A2	A3	A5	B10	A6	B4	B8	A8	B1	B2	A7	B3	A4
	X_2	1	1	1	1	0	1	0	1	0	0	1	1	0	1	0	0	0	0
Parent 2	X_1	A1	B5	B6	A5	B7	B4	B10	B9	A3	A2	B8	A6	A8	A7	B1	B2	A4	B3
	X_2	1	1	1	0	1	0	1	1	1	1	0	1	0	0	1	0	0	0

Table 7
Offspring disassembly sequence after the crossover.

Task ID		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Offspring 1	X_1	B5	B6	B7	A1	A5	B4	B10	B9	A3	A2	B8	A6	A8	B1	B2	A7	B3	A4
	X_2	1	1	1	1	0	0	1	1	1	1	0	1	0	1	0	0	0	0
Offspring 2	X_1	A1	B5	B6	B7	B9	A2	A3	A5	B10	A6	B4	B8	A8	B1	A7	B2	A4	B3
	X_2	1	1	1	1	0	1	0	1	0	0	1	1	0	1	0	0	0	0

Table 8
The individual sequence before and after mutation.

Task ID		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Parent	X_1	B5	B6	B7	B9	A1	B10	A5	B4	B1	A3	B8	A2	B3	A6	A8	A7	A4	B2
	X_2	1	1	1	1	1	0	1	1	1	1	0	1	0	1	0	0	0	0
Offspring	X_1	B5	B6	B7	B9	A1	B10	A5	B4	B8	B1	A3	A2	B3	A6	A8	A7	A4	B2
	X_2	1	1	1	1	1	0	1	1	0	1	1	1	0	1	0	0	0	0

4.4.1. Selection operation

The non-inferior solutions in the external file are all high-quality individuals in the multi-objective problem. In the algorithm iteration process, the non-inferior solutions in the external file are added to the population. Therefore, the individuals are selected randomly from the populations when performing the selection operator.

4.4.2. Two-point mapping crossover

The random crossover method easily produces infeasible solutions, which reduces the efficiency of the algorithm. For this reason, this paper uses a two-point mapping crossover to ensure that the crossover results are feasible solutions. Both the disassembly task sequence and the disassembly task variable use the two-point mapping crossover operation.

Taking the precedence matrix in Fig. 4 as an example, any two sequences are selected as parent individuals, as shown in Table 6. We randomly select two points in one parent as the intersection, such as the 4th and 15th positions; keep the sequence outside the two positions unchanged; and the sequence between the two positions is obtained by mapping another parent, as shown in X_1 in Table 7. The sequences inside and outside the intersection satisfy the precedence constraints, and the offspring must satisfy the precedence constraints. The offspring of the disassembly task variables are obtained using the same crossover method, as shown in X_2 in Table 7.

4.4.3. Single-point insertion mutation

Similarly, the random mutation operation is prone to producing infeasible solutions. Therefore, this paper adopts a single-point insertion mutation based on the precedence constraints to ensure the feasibility of the mutation results. We randomly select a point in the parent individual as the mutation point, such as task B8 at position 11 in the example given in Table 8; then find the immediate predecessor and the immediate successor closest to task B8, i.e., tasks B4 and B3; and the position where task B8 can be inserted includes before tasks B1 and A3 and after task A2. Next, we randomly select an insertion; and if task B8 is inserted before task B1, then the new individual is as shown in Table 8.

4.5. Simulated annealing algorithm

The simulated annealing algorithm has a strong local search ability. It accepts an inferior solution with a probability in the iterative process and finds better solutions through multiple cycles. This mechanism can avoid the algorithm falling into a local optimum. The specific process of the simulated annealing operation is as follows:

Step 1: Initialize all parameters, including the initial temperature, termination temperature, cooling coefficient, and the number of cycles at each temperature level. The current temperature is equal to the initial temperature.

Step 2: A new initial solution is constructed, and the optimal solution is recorded.

Step 3: A neighborhood solution is generated according to the two-point swapping operation [40].

Step 4: Compare the neighborhood solution with the current solution. If the neighborhood solution dominates the current solution, the current solution is replaced by the neighborhood solution. If the current solution dominates the neighborhood solution, the neighborhood solution is accepted according to the Metropolis criterion. If neither solution dominates the other, both solutions are retained.

Step 5: Update the global optimal solution.

Step 6: If the maximum number of cycles is not met, steps 3–5 are repeated.

Step 7: If the current temperature is greater than the minimum temperature, the temperature is lowered according to the cooling coefficient, and steps 3–6 are repeated.

Step 8: Output the optimal solution.

4.6. Procedures of the proposed algorithm

Combining the genetic algorithm and simulated annealing can enhance the performance of the hybrid algorithm and achieve complementary advantages. In this paper, the results of genetic operations are simulated annealing operations. Based on the multi-objective characteristics of the problem, a multi-objective genetic simulated annealing algorithm is proposed to optimize the multi-objective SPPDLB. The specific procedures of the proposed algorithm are as follows:

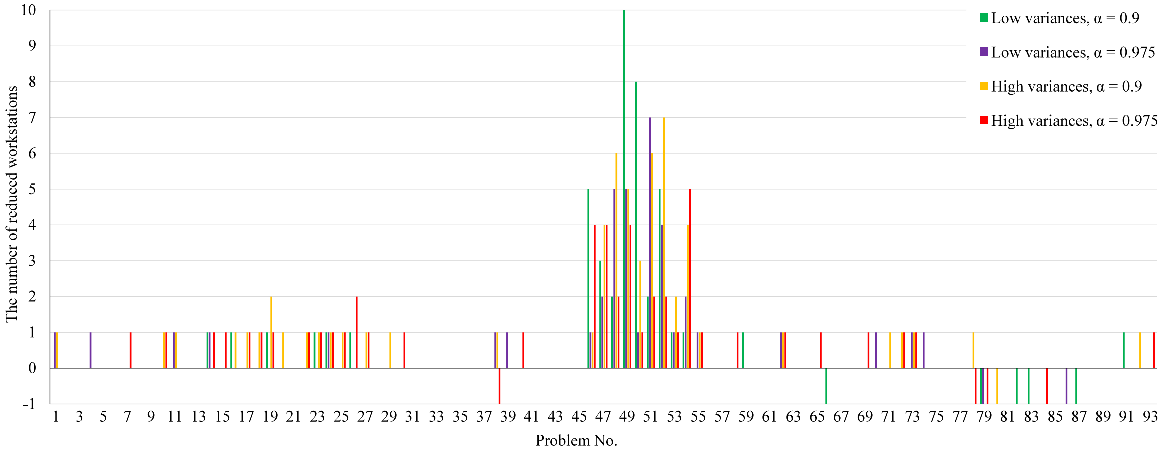


Fig. 7. Improvements of GSA in the number of workstations for SPALB.

Step 1: Parameter initialization. Set the number of individuals in the population, crossover probability (P_c), mutation probability (P_m), initial temperature (T_0) of simulated annealing, final temperature (T_{min}), cooling coefficient (τ), number of cycles at each temperature level (TI), and the external file $Q = \varnothing$.

Step 2: Population initialization. The initial solutions are generated according to the encoding operation, and the fitness value is calculated according to the decoding operation.

Step 3: Filter the non-inferior solutions according to the dominance relations and update the external file Q.

Step 4: Start the iteration. Set the current temperature of the iterations $T = T_0$.

Step 5: Perform the selection operation from the genetic operations in Section 4.4.1.

Step 6: If $rand > P_c$, perform the crossover operation from Section 4.4.2.

Step 7: If $rand > P_m$, perform the mutation operation from Section 4.4.3.

Step 8: Perform simulated annealing operations on the results of the genetic operations. First, a new neighborhood solution is generated, then the new solution is accepted according to the Metropolis criterion, and this operation is cycled TI times at each temperature level.

Step 9: Filter the non-inferior solutions according to the dominance relations and update the external file Q.

Step 10: If the number of non-inferior solutions in the external file is less than the number of individuals in the population, all the non-inferior solutions are added to the population, and new solutions are randomly generated and added to the population to keep the number of individuals in the population unchanged. Otherwise, a non-inferior solution with a large crowding distance is selected as the population.

Step 11: If $T < T_{min}$, then set $T = \tau \times T$, and go to Step 5; otherwise, go to Step 12.

Step 12: The iteration ends and the non-inferior solution in the external file is output.

5. Computational experiments and comparisons

This section will use the proposed genetic simulated annealing (GSA) to solve different types of parallel production line balancing problems, including 372 stochastic parallel assembly line balancing (SPALB) problems in Section 5.1, 45 parallel disassembly line balancing (PDLB) problems in Section 5.2, and a practical example of stochastic parallel partial disassembly line balancing (SPPDLB) in Section 5.3. These three types of problems are essentially the same. The performance of the proposed algorithm is verified by comparing the results of it with the results of different algorithms. The proposed algorithm is implemented in MATLAB 9.3 and runs on an Intel Core i5-8400 CPU 2.8 GHz computer with 16 GB RAM.

5.1. Comparison results of SPALB

The characteristics of the assembly line balancing problem are very similar to those of the disassembly line balancing problem. All tasks must be conducted on the assembly line, but only a part of the tasks are allowed to be conducted on the disassembly line. Therefore, complete line balancing is a special case of partial line balancing. Özcan [40] established a stochastic parallel assembly line balancing model, whose optimization objective is the number of workstations. The literature used the tabu search (TS) to test 372 examples under different parameters. These examples are parallel disassembly lines for the two products, with different numbers of tasks (N_1 and N_2), cycle times (CT_1 and CT_2), high and low variances, and different confidence levels (0.9 and 0.975). Each of the examples is independently solved 10 times using the proposed GSA, and the best-so-far solution of GSA is compared with the best-so-far solution of the TS. The results of the two algorithms are shown in Table 9.

The numbers and rates of GSA obtaining better solutions, the same solutions, and the worse solutions compared with TS in Table 9 are shown in Table 10. The results show that under different parameters, the rate of the same solutions in the two algorithms is the largest. The rates of GSA obtaining better solutions than TS are 18.28%, 22.56%, 33.33%, and 34.41%; and the rates of GSA obtaining worse solutions than TS are 5.38%, 2.15%, 1.08%, and 4.30%, respectively. Besides, the rates of better solutions are significantly larger than the rates of worse solutions, indicating that the optimization performance of GSA is better than that of TS.

Fig. 7 shows the improvement in the GSA results compared to the TS results. When the number of workstations obtained by the two algorithms is the same, the improvement is 0, which is not shown in the figure, and only the change in the number of workstations is shown in the figure. It can be seen from Fig. 7 that the reduction in the number of workstations is significantly more than the increase, and the maximum reduction is up to 10 workstations, but the increase is only one workstation, indicating that the GSA significantly improves the results of the SPALB cases.

Table 9
The results of TS and GSA for SPALB.

Problem	N_1	N_2	CT_1	CT_2	Low variances				High variances			
					$\alpha = 0.9$		$\alpha = 0.975$		$\alpha = 0.9$		$\alpha = 0.975$	
					TS	GSA	TS	GSA	TS	GSA	TS	GSA
Jaeschke–Jaeschke	9	9	10	14	8	8	10	**9**	11	**10**	13	13
			10	10	10	10	12	12	14	14	15	15
			18	10	7	7	8	8	9	9	11	11
Jackson–Jaeschke	11	9	10	14	9	9	11	**10**	11	11	13	13
			10	10	12	12	14	14	14	14	15	15
			21	18	5	5	6	6	6	6	7	7
Jackson–Jackson	11	11	10	13	11	11	12	12	11	11	14	**13**
			14	14	8	8	9	9	9	9	10	10
			21	14	7	7	7	7	7	7	8	8
Roszieg–Jackson	25	11	18	21	11	11	12	12	13	**12**	14	**13**
			21	21	10	10	11	**10**	11	**10**	12	12
			25	14	10	10	11	11	11	11	12	12
Roszieg–Roszieg	25	25	18	25	14	14	15	15	16	16	17	17
			21	21	15	**14**	16	**15**	16	16	18	**17**
			32	25	11	11	11	11	11	11	13	**12**
Sawyer–Roszieg	30	25	41	32	15	**14**	15	15	16	**15**	17	17
			47	25	14	14	15	15	16	**15**	18	**17**
			54	21	14	14	15	15	16	**15**	18	**17**
Sawyer–Sawyer	30	30	36	41	21	**20**	22	22	24	**22**	27	**26**
			36	36	22	22	24	24	25	**24**	28	28
			75	54	12	12	13	13	13	13	14	14
Gunther–Sawyer	35	30	61	75	15	15	16	16	17	**16**	19	**18**
			69	54	16	**15**	17	17	18	**17**	20	**19**
			81	36	19	**18**	20	**19**	21	**20**	24	**23**
Gunther–Gunther	35	35	61	69	19	19	20	20	22	**21**	25	**24**
			69	69	18	**17**	19	19	20	20	24	**22**
			81	61	17	17	18	18	20	**19**	23	**22**
Kilbridge–Gunther	45	35	79	81	15	15	16	16	17	17	19	19
			69	69	18	18	19	19	20	**19**	22	22
			184	61	13	13	14	14	15	15	17	**16**
Kilbridge–Kilbridge	45	45	79	184	12	12	12	12	12	12	13	13
			92	92	14	14	15	15	15	15	16	16
			138	110	10	10	11	11	11	11	12	12
Hahn–Kilbridge	53	45	2338	92	14	14	15	15	15	15	16	16
			2004	69	18	18	19	19	19	19	21	21
			2338	184	10	10	11	11	11	11	12	12
Hahn–Hahn	53	53	2004	4676	12	12	13	13	13	13	14	14
			2806	2806	12	12	13	**12**	13	**12**	14	*15*
			4676	3507	8	8	9	**8**	9	9	9	9
Tonge–Hahn	70	53	293	2004	22	22	24	24	25	25	28	**27**
			410	4676	13	13	14	14	14	14	16	16
			468	3507	13	13	14	14	14	14	16	16
Tonge–Tonge	70	70	364	410	21	21	22	22	23	23	26	26
			468	468	17	17	18	18	19	19	21	21
			527	293	22	22	23	23	24	24	27	27
Wee–Mag–Tonge	75	70	50	320	55	**50**	63	**62**	63	**62**	71	**67**
			52	364	48	**45**	57	**55**	60	**56**	66	**62**
			54	527	42	**40**	49	**44**	54	**48**	58	**56**
Wee–Mag–Wee–Mag	75	75	50	56	77	**67**	95	**90**	103	**98**	113	**109**
			52	52	82	**74**	104	**103**	107	**104**	113	**112**
			56	54	67	**65**	83	**76**	97	**91**	108	**106**
Arcus83–Wee–Mag	83	75	5048	50	59	**54**	67	**63**	70	**63**	74	**72**
			5408	54	50	**49**	56	**55**	62	**60**	70	**69**
			5853	56	47	**46**	51	**49**	58	**54**	66	**61**
Arcus83–Arcus83	83	83	5048	5408	34	34	36	**35**	38	**37**	43	**42**
			6883	6883	25	25	26	26	28	28	31	31
			8898	6309	23	23	24	24	26	26	29	29
Lutz3–Arcus83	89	83	110	6309	31	31	33	33	35	35	39	**38**
			127	7571	27	**26**	28	28	29	29	32	32
			150	8898	22	22	23	23	24	24	27	27

(*continued on next page*)

Table 9 (*continued*).

Problem	N_1	N_2	CT_1	CT_2	Low variances				High variances			
					$\alpha = 0.9$		$\alpha = 0.975$		$\alpha = 0.9$		$\alpha = 0.975$	
					TS	GSA	TS	GSA	TS	GSA	TS	GSA
Lutz3–Lutz3	89	89	110	150	30	30	32	32	33	33	37	37
			118	118	33	33	35	**34**	37	**36**	41	**40**
			137	127	29	29	31	31	32	32	36	36
Mukherje–Lutz3	94	89	301	137	30	30	32	32	33	33	37	37
			324	118	31	31	33	33	35	35	39	**38**
			351	150	26	*27*	28	28	29	29	32	32
Mukherje–Mukherje	94	94	301	301	33	33	35	35	36	36	40	40
			301	351	30	30	32	32	33	33	37	37
			351	324	29	29	31	31	32	32	36	**35**
Arcus111–Mukherje	111	94	8847	301	36	36	39	**38**	40	40	45	45
			9400	324	34	34	36	36	38	**37**	42	42
			10027	351	31	31	33	33	35	**34**	39	**38**
Arcus111–Arcus111	111	111	8847	9400	39	39	42	**41**	44	**43**	49	**48**
			11378	11378	31	31	33	**32**	33	33	37	37
			17067	10743	26	26	28	28	29	29	32	32
Bartholdi–Arcus111	148	111	564	11378	26	26	28	28	29	29	31	31
			705	11570	24	24	25	25	26	26	28	28
			805	7571	31	31	33	33	35	**34**	37	*38*
Bartholdi–Bartholdi	148	148	513	564	23	*24*	24	*25*	25	25	27	*28*
			626	626	20	20	21	21	21	*22*	23	23
			805	705	17	17	17	17	18	18	19	19
Lee–Bartholdi	205	148	1510	564	28	*29*	30	30	31	31	34	34
			2077	626	22	*23*	23	23	24	24	26	26
			2832	705	18	18	19	19	19	19	20	*21*
Lee–Lee	205	205	1699	2643	25	25	26	26	27	27	29	29
			2266	2266	23	23	23	*24*	24	24	26	26
			2832	2454	19	*20*	20	20	21	21	22	22
Scholl–Lee	297	205	1935	2831	50	50	52	52	54	54	60	60
			2247	1699	50	50	53	53	54	54	60	60
			2787	1510	45	45	47	47	49	49	53	53
Scholl–Scholl	297	297	2049	2680	68	**67**	71	71	73	73	81	81
			2111	2111	75	75	78	78	82	**81**	90	90
			2787	2247	63	63	66	66	68	68	75	**74**

Table 10
The statistical results of GSA for SPALB.

Data statistics	Low variances		High variances	
	$\alpha = 0.9$	$\alpha = 0.975$	$\alpha = 0.9$	$\alpha = 0.975$
The number of better solutions	17	21	31	32
The number of identical solutions	71	70	61	57
The number of worse solutions	5	2	1	4
The rate of better solutions	18.28%	22.58%	33.33%	34.41%
The rate of identical solutions	76.34%	75.27%	65.59%	61.29%
The rate of worse solutions	5.38%	2.15%	1.08%	4.30%

5.2. Comparison results of PDLB

This section takes 45 basic data in the Ref. [40] as examples to construct parallel complete disassembly line balancing with a deterministic disassembly time. There are two optimization objectives, namely, the number of stations (*NS*) and the smoothness index (*SI*). The optimization methods include the CPLEX solver, the genetic algorithm (GA), simulated annealing (SA), and the proposed GSA, the purpose of which is to compare the proposed hybrid algorithm with the exact method and two basic algorithms.

In the PDLB problems, *SI* is essentially the same as *NS*, and *NS* can be compared to the lower bound (*LB*). The solution obtained by the meta-heuristic method can directly obtain the two objectives. Since *SI* is a non-linear objective, *NS* is optimized first, and then *SI* is optimized in CPLEX. The maximum running time of CPLEX is 3600 s. In the nine small-scale parallel disassembly line balancing problems, the CPLEX solver can quickly find the optimal solution, and its solution time (CPU) and gaps of the two objectives (namely, Gap1 and Gap2) are recorded. The *NS* and *SI* obtained by CPLEX are the termination criteria of the GA, SA, and GSA; and the maximum running time is less than 3600 s. The solution and running time obtained by each algorithm are recorded, as shown in Table 11.

Analysis Table 11 shows that both SA and GSA have obtained the optimal solutions of *NS* and *SI* for all problems, the GA has obtained the optimal solution of *NS* for all problems, and the GA has three problems in which it cannot find the optimal solution of *SI*. The number of problems in which CPLEX, the GA, SA, and GSA obtain the shortest solution time is 0, 2, 4, and 3, respectively, which indicates that the performance of SA and GSA is better than that of CPLEX and the GA.

In 36 medium-sized parallel disassembly line balancing problems, CPLEX could not find the optimal solution in 3600 s; and the

Table 11
Results of CPLEX, GA, SA, and GSA for small-scale PDLB.

No.	N_1	N_2	CT_1	CT_2	*LB*	CPLEX					GA			SA			GSA		
						NS	Gap1 (%)	*SI*	Gap2 (%)	CPU	*NS*	*SI*	CPU	*NS*	*SI*	CPU	*NS*	*SI*	CPU
1	9	9	10	14	7	7	0.00	324	17.45	1.19	7	*336*	3600	7	324	0.16	7	324	**0.13**
2	9	9	10	10	8	8	0.00	6	0.36	0.93	8	6	0.60	8	6	**0.01**	8	6	0.07
3	9	9	18	10	6	6	0.00	106	0.00	0.35	6	106	0.85	6	106	**0.08**	6	106	1.01
4	11	9	10	14	8	8	0.00	369	1.89	10.59	8	*371*	3600	8	369	92.40	8	369	**5.90**
5	11	9	10	10	9	9	0.00	7	1.29	12.27	9	7	0.66	9	7	**0.01**	9	7	0.03
6	11	9	21	18	5	5	0.00	1809	4.16	7.22	5	1809	**0.89**	5	1809	11.56	5	1809	**0.89**
7	11	11	10	13	9	9	0.00	1422	1.07	81.39	9	*1510*	3600	9	1422	**7.27**	9	1422	16.13
8	11	11	14	14	7	7	0.00	6	0.14	51.55	7	6	0.78	7	6	**0.01**	7	6	0.03
9	11	11	21	14	6	6	0.00	82	1.02	52.58	6	82	**1.14**	6	82	9.81	6	82	9.41

Table 12
Results of CPLEX, GA, SA, and GSA for medium-scale PDLB.

No.	n_1	n_2	CT_1	CT_2	*LB*	CPLEX				GA		SA		GSA	
						NS	Gap1 (%)	*SI*	Gap2 (%)	*NS*	*SI*	*NS*	*SI*	*NS*	*SI*
10	25	11	18	21	10	10	0.00	1227	5.82	10	1251	10	**1225**	10	**1225**
11	25	11	21	21	9	9	0.00	38	6.85	9	**36**	9	**36**	9	**36**
12	25	11	25	14	9	9	0.00	6976	11.54	9	6972	9	**6960**	9	**6960**
13	25	25	18	25	12	13	0.00	18197	22.28	13	17847	13	**17613**	13	**17613**
14	25	25	21	21	12	12	0.00	**2**	16.38	13	45	13	45	12	**2**
15	25	25	32	25	9	9	0.00	**765**	10.41	10	77865	10	77417	9	815
16	30	25	41	32	12	12	0.00	6345	21.40	13	191149	13	189601	12	**5357**
17	30	25	47	25	12	12	7.69	1891	21.42	13	130524	13	130524	12	**1457**
18	30	25	54	21	12	12	7.69	**48**	25.57	13	12174	13	12170	12	82
19	30	30	36	41	17	17	5.56	*	*	18	171916	18	166438	17	**3330**
20	30	30	36	36	18	19	5.26	*	*	20	268	19	**84**	19	**84**
21	30	30	75	54	11	11	0.00	81318	29.06	11	76796	11	76700	11	**76644**
22	35	30	61	75	13	13	0.00	1022530	22.89	13	954432	13	**939356**	13	941472
23	35	30	69	54	13	14	7.14	*	*	14	**110656**	14	110936	14	111434
24	35	30	81	36	15	16	6.25	*	*	16	7586	16	7358	16	**7194**
25	35	35	61	69	15	16	6.25	1956064	27.94	16	1507114	16	**1332862**	16	1511946
26	35	35	69	69	14	15	6.67	405	24.30	15	353	15	385	15	**341**
27	35	35	81	61	14	14	0.00	35264	22.50	15	2330195	15	2048083	14	**29916**
28	45	35	79	81	13	13	7.14	20902	30.64	14	3242559	14	3276581	13	**14392**
29	45	35	69	69	15	15	6.25	*	*	16	301	16	303	15	**0**
30	45	35	184	61	11	11	0.00	100714	21.91	12	12322920	12	12327736	11	**79370**
31	45	45	79	184	10	10	9.09	*	*	11	19729232	11	19708586	10	**3924**
32	45	45	92	92	12	12	0.00	*	*	13	652	13	664	12	**0**
33	45	45	138	110	10	10	0.00	6213428	30.23	10	5557308	10	5555302	10	**5553274**
34	53	45	2338	92	12	13	7.69	*	*	13	903220442	13	910216574	13	**899665372**
35	53	45	2004	69	15	16	6.25	*	*	16	140340230	16	**136311138**	16	139162956
36	53	45	2338	184	9	10	10.00	*	*	10	4649729994	10	4805366892	10	**4644412598**
37	53	53	2004	4676	10	11	9.09	24905144	20.19	11	18241160	11	18225088	11	**17954796**
38	53	53	2806	2806	10	11	9.09	837258	23.03	11	725568	11	722624	11	**722166**
39	53	53	4676	3507	7	7	12.50	*	*	8	24662630	8	24652292	7	**50**
40	70	53	293	2004	19	20	5.00	*	*	20	20314711030	20	18207960416	20	**18196017406**
41	70	53	410	4676	12	12	0.00	*	*	12	**14799525510**	12	14805035850	12	14823659296
42	70	53	468	3507	12	12	0.00	*	*	12	6260101754	12	6267061422	12	**6251617182**
43	70	70	364	410	19	19	0.00	*	*	19	186403734	19	186432778	19	**186344100**
44	70	70	468	468	15	16	6.25	*	*	16	13696	16	13712	16	**13692**
45	70	70	527	293	19	19	0.00	*	*	20	2207356064	20	2214620612	19	**197137009**

termination criteria of the GA, SA, and GSA are set as a running time of 3600 s. The results of the four methods are shown in Table 12, where * indicates that CPLEX cannot obtain a feasible solution within 3600 s. If the number of stations obtained by different methods is the same as the lower bound, it can be considered the optimal solution. The best-so-far solutions are marked in the table, and the number of best-so-far solutions obtained by GSA is greater than those of the other three methods.

Combining Tables 11 and 12, the number of best-so-far solutions for the 45 problems is calculated for the four methods, as shown in Table 13. On the *NS* objective, GSA and CPLEX obtain all the best-so-far solutions for 45 problems while the rates of the GA and SA obtaining the best-so-far solution are only 66.7% and 68.9%, respectively. On the *SI* objective, the rates of CPLEX, the GA, SA, and GSA obtaining the best-so-far solutions are 36.7%, 20.0%, 37.8%, and 84.4%, respectively. The performance of GSA is better than that of the exact method and the basic algorithms in a given computational time.

5.3. Comparison results of SPPDLB

In this section, we construct stochastic parallel partial disassembly line balancing (SPPDLB) with waste television and refrigerator lines in a disassembly enterprise in China, and the proposed multi-objective genetic simulated annealing (MOGSA) is used to optimize the number of stations, smoothness index, and disassembly profits of the parallel disassembly lines. The waste televisions and refrigerators contain 27 and 25 parts, respectively. The specific information of the parts is shown in Appendix Tables A.1 and A.2. The precedence relationships are shown in Appendix Figs. A.1 and A.2.

The cycle times of the waste television and refrigerator lines are set to $CT_1 = \{68, 76\}$ and $CT_2 = \{88, 94\}$, respectively. The confidence level for stochastic disassembly times is set to $\alpha =$

Table 13
The statistical results of CPLEX, GA, SA, and GSA for PDLB.

Data statistics	*NS*				*SI*			
	CPLEX	GA	SA	GSA	CPLEX	GA	SA	GSA
The number of better solutions	45	30	31	45	12	9	17	38
The rate of better solutions	100%	66.7%	68.9%	100%	36.7%	20.0%	37.8%	84.4%

Table 14
The main parameters of SPPDLB.

No.	α	$\Phi^{-1}(\alpha)$	CT_1	CT_2	No.	α	$\Phi^{-1}(\alpha)$	CT_1	CT_2	No.	α	$\Phi^{-1}(\alpha)$	CT_1	CT_2	No.	α	$\Phi^{-1}(\alpha)$	CT_1	CT_2
1	0.9	1.2816	68	88	4	0.9	1.2816	76	94	7	0.95	1.6449	76	88	10	0.975	1.9600	68	94
2	0.9	1.2816	68	94	5	0.95	1.6449	68	88	8	0.95	1.6449	76	94	11	0.975	1.9600	76	88
3	0.9	1.2816	76	88	6	0.95	1.6449	68	94	9	0.975	1.9600	68	88	12	0.975	1.9600	76	94

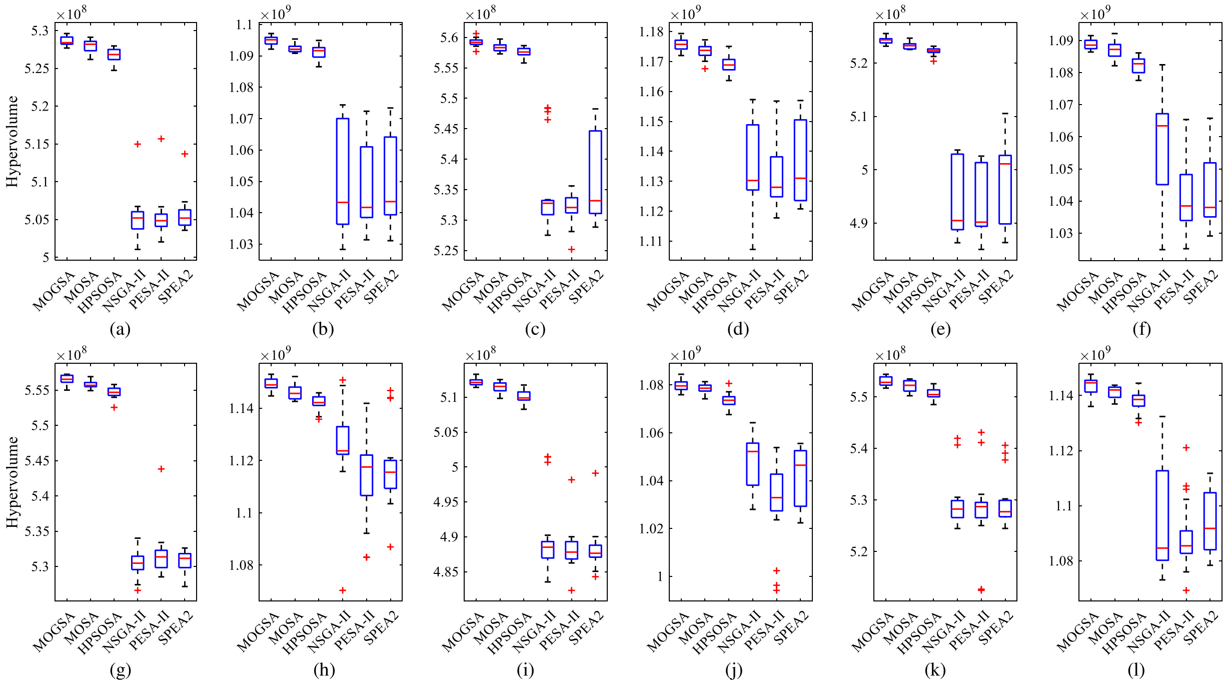


Fig. 8. Box plots of six algorithms on the hypervolume indicator.

{0.9, 0.95, 0.975}. According to different parameter combinations, there are 12 parameter conditions, as shown in Table 14.

The comparison algorithms in this section include multi-objective simulated annealing (MOSA) [55], hybrid particle swarm optimization integrated with simulated annealing (HPSOSA) [56], the non-dominated sorting genetic algorithm II (NSGA-II) [54], the Pareto envelope-based selection algorithm II (PESA-II) [57], and the strength Pareto evolutionary algorithm 2 (SPEA2) [58]. MOGSA, MOSA, and HPSOSA are multi-objective algorithms with local search strategies while NSGA-II, PESA-II, and SPEA2 are general multi-objective algorithms. The MOGSA is compared with the five algorithms to verify the performance of the global search and local search of the proposed algorithm. The population for each algorithm is set to 50, and the termination condition is 100 s of running time. The crossover and mutation probabilities related to the genetic operations are set to 0.8 and 0.2, respectively. The initial temperature, the number of cycles, and the cooling rate in the simulated annealing operation are set to 200, 10, and 0.974, respectively. The inertia weight and two learning factors in HPSOSA are set to 0.2, 0.8, and 0.9, respectively. The number of grids per dimension in the PESA-II is set to 10. The other parameters in the model are set as follows: $\rho_0 = 0.2$, $C_f = 0.1$, $C_g = 0.1$, and $C_s = 0.01$. The hypervolume [59] and epsilon [60] are used to evaluate the solutions obtained by the algorithms. The reference point for the hypervolume is set to (52, 10000, 0). Each algorithm runs 20 times independently under different parameters. The hypervolume and epsilon values of each algorithm under different parameter conditions are recorded, and the box plots are shown in Figs. 8 and 9, respectively.

The smaller the hypervolume is, the closer the non-inferior solution is to the true Pareto front. The gap between the average levels of MOGSA, MOSA, and HPSOSA is small, but the average levels of MOGSA, MOSA, and HPSOSA are significantly better than those of the NSGA-II, PESA-II, and SPEA2, indicating that the local search strategy can improve the quality of the solution. MOGSA only has outliers in condition 3, and no outliers appeared under other conditions, especially in conditions 8, 9, and 11. In addition, the numbers of outliers of MOGSA, MOSA, and HPSOSA are less than those of the NSGA-II, PESA-II, and SPEA2. Larger values of epsilon are preferred. Similarly, the stability and the average level

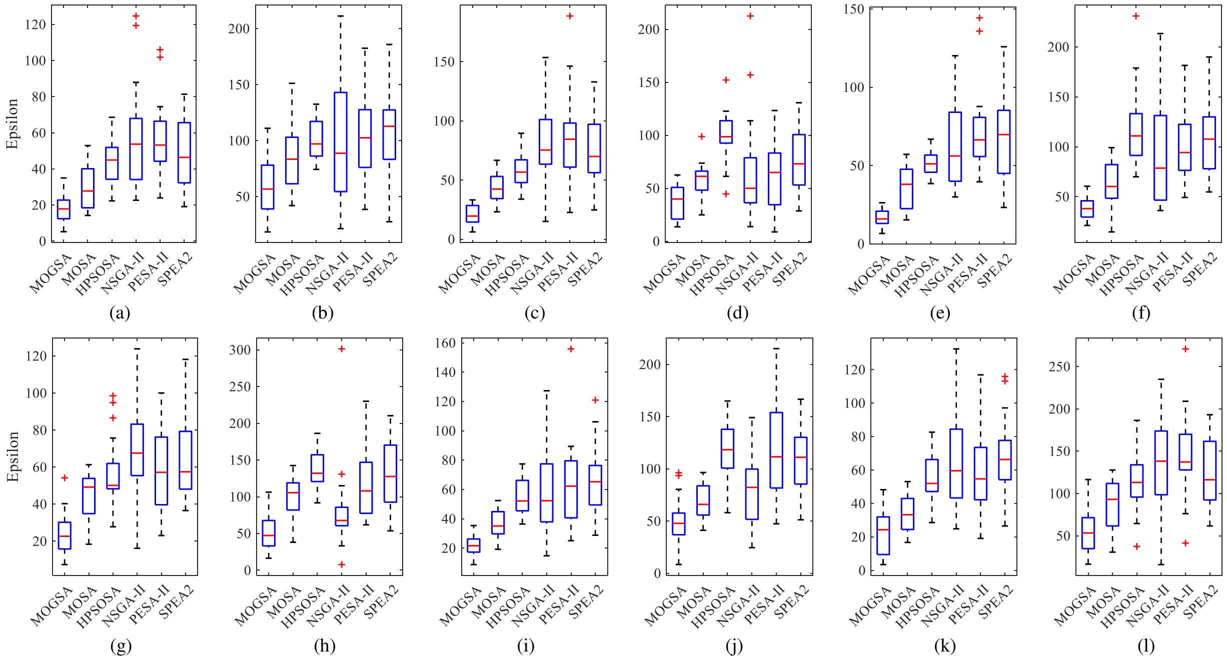


Fig. 9. Box plots of six algorithms on the epsilon indicator.

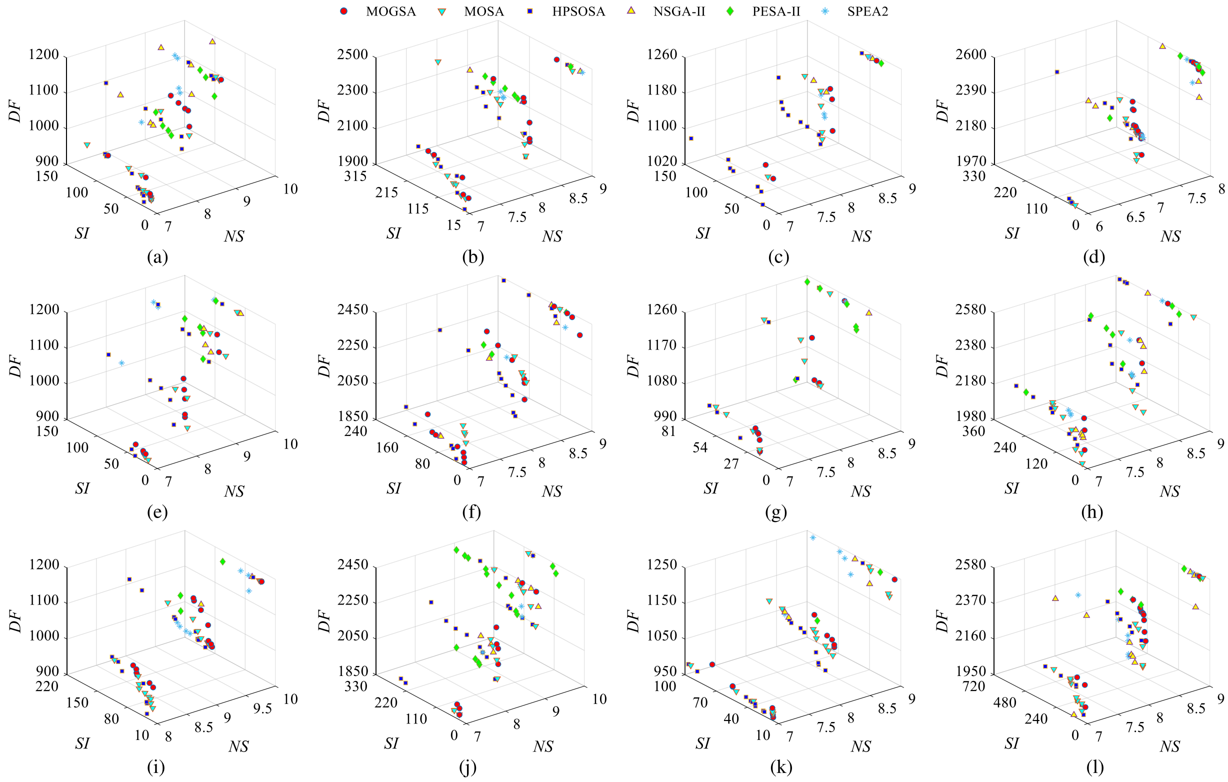


Fig. 10. Pareto front of five algorithms under different parameters.

of MOGSA are better than those of MOSA, HPSOSA, NSGA-II, PESA-II, and SPEA2 on the epsilon index. The number of outliers of all algorithms under different parameters is small. The boxplot comparisons of the two indicators show that MOGSA performs better than MOSA, HPSOSA, NSGA-II, PESA-II, and SPEA2.

In addition, Tables 15 and 16 record the minimum (Min), average (Avg), maximum (Max), and standard deviation (SD) of the hypervolume and epsilon values obtained by the six algorithms under 12 parameter conditions. Table 15 shows that on the four indexes of the hypervolume in the 12 problems, the numbers

Table 15
Statistical results of five algorithms on the hypervolume index. (Unit: 10^8)

$\Phi^{-1}(\alpha)$		1.2816				1.6449				1.9600			
CT_1		68	68	76	76	68	68	76	76	68	68	76	76
CT_2		88	94	88	94	88	94	88	94	88	94	88	94
Min	MOGSA	**52.77**	**109.22**	**55.77**	**117.20**	**52.32**	**108.64**	**55.50**	**114.47**	**51.14**	**107.59**	**55.17**	113.60
	MOSA	52.62	109.08	55.73	116.76	52.25	108.21	55.49	114.26	50.98	107.42	55.02	**113.69**
	HPSOSA	52.47	108.65	55.58	116.37	52.03	107.76	55.25	113.58	50.83	106.76	54.85	113.01
	NSGA-II	50.10	102.83	52.75	110.72	48.63	102.48	52.66	107.03	48.35	102.80	52.45	107.31
	PESA-II	50.20	103.14	52.52	111.78	48.51	102.51	52.85	108.30	48.23	99.41	51.25	106.93
	SPEA2	50.36	103.11	52.88	112.08	48.64	102.91	52.72	108.70	48.43	102.24	52.45	107.84
Avg	MOGSA	**52.86**	**109.48**	**55.92**	**117.57**	**52.42**	**108.88**	**55.65**	**114.91**	**51.21**	**107.98**	**55.30**	**114.35**
	MOSA	52.80	109.25	55.83	117.34	52.32	108.73	55.58	114.62	51.15	107.85	55.21	114.10
	HPSOSA	52.67	109.12	55.75	116.87	52.22	108.22	55.47	114.22	51.01	107.35	55.06	113.79
	NSGA-II	50.50	105.19	53.48	113.48	49.51	105.63	53.04	112.58	48.97	104.87	52.92	109.29
	PESA-II	50.52	104.74	53.20	113.21	49.40	104.18	53.16	111.35	48.82	103.13	52.82	108.85
	SPEA2	50.57	105.07	53.61	113.69	49.81	104.34	53.07	111.70	48.82	104.19	52.93	109.35
Max	MOGSA	**52.96**	**109.71**	**56.06**	**117.94**	**52.55**	109.15	**55.72**	**115.30**	**51.33**	**108.45**	**55.44**	**114.77**
	MOSA	52.91	109.54	55.97	117.72	52.47	**109.22**	55.69	115.21	51.25	108.14	55.35	114.37
	HPSOSA	52.80	109.49	55.86	117.51	52.32	108.61	55.58	114.59	51.17	108.06	55.25	114.45
	NSGA-II	51.50	107.44	54.84	115.73	50.37	108.24	53.40	115.08	50.15	106.42	54.20	113.24
	PESA-II	51.57	107.24	53.56	115.68	50.26	106.54	54.38	114.19	49.82	105.38	54.31	112.11
	SPEA2	51.37	107.34	54.83	115.70	51.05	106.58	53.26	114.68	49.91	105.55	54.06	111.18
SD	MOGSA	**0.06**	0.15	**0.06**	**0.21**	**0.06**	**0.16**	0.06	**0.22**	**0.05**	0.22	**0.08**	0.31
	MOSA	0.08	**0.13**	0.06	0.23	**0.06**	0.26	**0.04**	0.29	0.07	**0.19**	0.11	**0.24**
	HPSOSA	0.09	0.21	0.08	0.30	0.07	0.25	0.08	0.27	0.09	0.28	0.10	0.32
	NSGA-II	0.29	1.69	0.68	1.43	0.73	1.68	0.18	1.66	0.52	1.12	0.45	1.86
	PESA-II	0.28	1.30	0.26	1.16	0.65	1.21	0.32	1.44	0.29	1.70	0.70	1.22
	SPEA2	0.22	1.45	0.69	1.37	0.71	1.13	0.15	1.41	0.29	1.14	0.45	1.09

Table 16
Statistical results of five algorithms on the epsilon index.

$\Phi^{-1}(\alpha)$		1.2816				1.6449				1.9600			
CT_1		68	68	76	76	68	68	76	76	68	68	76	76
CT_2		88	94	88	94	88	94	88	94	88	94	88	94
Min	MOGSA	**4.93**	**25.14**	**6.28**	13.70	**5.01**	17.14	**7.41**	16.40	**11.64**	**8.40**	**3.51**	18.88
	MOSA	14.26	41.97	23.27	24.95	15.33	**14.73**	18.39	38.11	19.20	41.14	16.76	30.70
	HPSOSA	22.31	74.30	33.96	44.70	38.51	69.97	27.83	91.72	36.43	58.01	28.62	37.43
	NSGA-II	22.65	21.44	15.15	13.86	29.98	36.14	16.19	**7.75**	14.76	24.53	24.88	**15.98**
	PESA-II	23.94	38.55	22.74	**8.88**	39.62	49.28	23.07	61.81	25.10	47.37	19.17	41.36
	SPEA2	19.14	27.55	24.85	28.68	23.29	54.77	36.56	53.67	28.82	51.19	26.57	61.62
Avg	MOGSA	**17.81**	**59.59**	**20.87**	**38.04**	**13.49**	**32.83**	**23.85**	**54.60**	**22.95**	**49.34**	**24.35**	**58.39**
	MOSA	29.22	84.92	42.46	58.07	36.41	61.49	44.71	97.88	35.86	68.71	34.17	86.46
	HPSOSA	44.39	100.92	58.29	100.13	51.62	119.44	57.59	138.41	54.57	118.31	55.34	114.55
	NSGA-II	57.22	97.15	78.36	66.14	63.89	92.84	68.80	79.76	58.94	77.46	65.94	136.54
	PESA-II	56.39	104.98	85.52	61.42	71.64	103.64	58.67	120.54	64.25	121.69	60.60	145.81
	SPEA2	49.15	107.92	77.70	76.21	67.48	109.55	65.15	130.24	65.10	108.46	68.80	126.44
Max	MOGSA	**34.99**	**111.04**	**33.30**	**62.56**	**24.44**	**55.53**	**54.22**	**106.30**	**38.28**	**96.23**	**48.24**	**117.39**
	MOSA	52.93	151.14	66.71	98.85	57.18	99.12	61.33	142.49	52.43	96.47	53.10	127.79
	HPSOSA	68.58	132.54	89.54	152.18	66.87	231.06	98.55	186.33	77.41	164.94	82.67	186.46
	NSGA-II	124.70	211.28	153.50	212.55	120.11	213.43	123.87	301.35	127.31	149.08	132.39	234.98
	PESA-II	106.01	182.52	188.55	123.36	144.30	181.54	100.03	230.03	155.84	215.15	116.88	270.91
	SPEA2	81.40	185.86	132.88	130.72	125.86	189.94	118.18	210.30	120.95	166.65	115.93	193.11
SD	MOGSA	**8.09**	26.06	**8.33**	**16.34**	**5.34**	**10.37**	**11.47**	28.16	**7.06**	23.71	14.48	**25.93**
	MOSA	12.21	28.95	12.47	16.90	14.09	23.73	11.87	29.78	9.47	**16.59**	**11.43**	31.96
	HPSOSA	12.51	**19.70**	15.04	23.72	8.11	37.13	18.22	**23.83**	12.15	27.11	15.30	34.41
	NSGA-II	28.65	54.54	36.14	49.07	28.61	53.31	26.60	59.19	32.21	33.01	29.51	58.83
	PESA-II	21.28	37.31	41.31	33.12	27.41	39.53	23.44	53.97	29.23	49.52	27.10	49.79
	SPEA2	19.64	40.65	28.23	30.65	28.30	37.97	22.73	46.09	22.94	31.86	22.53	40.73

of times that MOGSA obtains the best-so-far indicators are 11, 12, 11, and 8, respectively. Second, MOSA obtains the best-so-far value 1, 0, 1, and 5 times the corresponding index; while the other four algorithms do not have the best-so-far value on any index in any problem. Similarly, Table 16 shows that on the four indexes of epsilon in the 12 problems, the numbers of times that MOGSA obtains the best-so-far indicators are 8, 12, 12, and 8, respectively. MOSA, HPSOSA, NSGA-II, and PESA-II obtain the best-so-far value a total of 4, 0, 0, and 4 times the corresponding indicators, respectively, while the SPEA2 does not obtain the best-so-far value on any known index. The results of the two tables show that MOGSA is superior to MOSA, HPSOSA, NSGA-II, PESA-II, and SPEA2 on both hypervolume and epsilon indicators under different parameter conditions.

The Pareto front when the six algorithms obtain the maximum hypervolume value under different parameters is shown in

Table 17
Evaluation of non-inferior solution sets of five algorithms.

No.		1	2	3	4	5	6	7	8	9	10	11	12
LB		6	6	6	6	7	6	6	6	7	6	6	6
$NS_{\min}$	MOGSA	7	7	7	**6**	**7**	7	7	7	8	7	7	7
	MOSA	7	7	7	**6**	**7**	7	7	7	8	7	7	7
	HPSOSA	7	7	7	**6**	**7**	7	7	7	8	7	7	7
	NSGA-II	8	8	8	7	9	7	9	7	9	8	8	7
	PESA-II	8	8	9	7	9	8	8	7	9	8	8	8
	SPEA2	8	8	8	7	8	8	9	7	9	8	8	8
N_{nis}	MOGSA	11	12	6	**14**	11	**17**	9	6	14	9	13	13
	MOSA	12	**16**	7	9	8	9	**12**	14	**17**	13	17	14
	HPSOSA	**16**	13	**16**	8	**12**	**17**	6	**16**	16	**19**	**18**	**15**
	NSGA-II	7	4	3	9	4	5	1	8	2	6	4	10
	PESA-II	7	6	1	4	5	3	6	7	3	15	2	5
	SPEA2	5	4	4	5	4	3	1	7	7	6	5	7
GD	MOGSA	**0.00**	**0.04**	**0.00**	**0.03**	**0.03**	0.09	0.06	0.33	**0.01**	**0.00**	**0.00**	**0.01**
	MOSA	0.05	0.10	0.20	0.15	0.23	**0.00**	0.11	**0.29**	0.11	0.22	0.07	0.09
	HPSOSA	0.10	0.08	0.41	0.80	0.73	0.17	0.25	0.75	0.30	0.38	0.08	0.11
	NSGA-II	0.32	0.15	0.20	0.56	0.51	0.20	**0.00**	0.45	0.24	0.14	0.26	0.33
	PESA-II	0.14	0.09	0.00	0.53	0.93	0.10	0.33	0.85	0.61	0.52	0.12	0.10
	SPEA2	0.38	0.12	0.22	0.17	2.18	0.09	0.54	0.45	0.26	0.45	0.36	0.30
IGD	MOGSA	**0.03**	**0.04**	**0.04**	**0.03**	**0.05**	**0.03**	0.09	**0.10**	**0.01**	**0.03**	**0.00**	**0.04**
	MOSA	0.05	0.06	0.17	0.07	0.09	0.06	**0.05**	0.19	0.06	0.06	0.04	0.08
	HPSOSA	0.06	0.07	0.30	0.14	0.26	0.10	0.14	0.22	0.11	0.10	0.08	0.07
	NSGA-II	0.22	0.20	0.24	0.08	0.38	0.15	0.32	0.11	0.22	0.13	0.21	0.15
	PESA-II	0.16	0.18	0.37	0.19	0.51	0.18	0.20	0.56	0.34	0.35	0.17	0.13
	SPEA2	0.28	0.20	0.22	0.14	1.16	0.14	0.38	0.31	0.23	0.28	0.22	0.18
NR	MOGSA	**0.73**	**0.85**	**0.86**	**0.39**	**0.67**	0.38	**0.67**	**0.50**	**0.86**	**0.82**	**0.79**	**0.77**
	MOSA	0.27	0.00	0.00	0.11	0.33	**0.56**	0.25	0.13	0.07	0.00	0.14	0.08
	HPSOSA	0.00	0.08	0.00	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	NSGA-II	0.00	0.00	0.00	0.11	0.00	0.00	0.08	0.38	0.07	0.18	0.00	0.00
	PESA-II	0.00	0.00	0.14	0.11	0.00	0.06	0.00	0.00	0.00	0.00	0.07	0.15
	SPEA2	0.00	0.08	0.00	0.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 18
Two disassembly balancing schemes for SPPDLB.

No.	*NS*	*SI*	*DF*	n_1	n_2	*AT*	*m*	Sequence	T_m	Δ (%)
S_1	7	9.91	929.88	16	19	1449	1	B1 → B2 → B14 → A1	1442.02	99.52
							2	B8 → B10 → A2	1443.79	99.64
							3	B9 → B20 → B11 → B12 → B15 → B16 → B18 → B19 → B21	1448.70	99.98
							4	A5 → A15 → B17 → A10 → A19	1444.86	99.71
							5	A7 → A8 → A6 → A9 → A16	1448.23	99.95
							6	B13 → B23 → B24	1447.11	99.87
							7	A14 → A21 → A22 → B22 → A4 → A3	1448.02	99.93
S_2	9	24.51	1183.1	26	25	1449	1	B1 → B2 → B14 → A1	1442.02	99.52
							2	B8 → B10 → A2	1443.79	99.64
							3	B9 → B20 → B11 → B12 → B15 → B16 → B18 → B19 → B21	1448.70	99.98
							4	A5 → A15 → B17 → A10 → A19	1444.86	99.71
							5	A7 → A8 → A6 → A9 → A16	1448.23	99.95
							6	B13 → B23 → B24	1447.11	99.87
							7	A14 → A21 → A22 → B22 → A18 → A13	1434.48	99.00
							8	B3 → B4 → A17 → B5 → B6 → B7 → A24 → A26 → A4	1441.45	99.48
							9	A20 → A25 → A11 → A12 → A23 → B25 → A27	1433.66	98.94

Fig. 10. MOGSA is closer to the Pareto front than the other algorithms. Moreover, the minimum number of stations ($NS_{\min}$), the number of non-inferior solutions (N_{nis}), the generational distance (GD), the inverted generational distance (IGD), and the dominance rate (DR) obtained by the six algorithms are calculated, as shown in Table 17. The analysis shows that MOGSA, MOSA, and HPSOSA obtain lower bounds for three of the 12 problems, while other algorithms do not obtain any lower bounds. Furthermore, the algorithms that obtain the maximum $NS_{\min}$ include MOGSA, MOSA, and HPSOSA. The numbers of times that MOGSA obtains the minimum GD, the minimum IGD, and the maximum DR on the 12 problems are 9, 11, and 11, respectively; while the evaluation indexes of the other algorithms are worse than those of MOGSA. It can be seen that the quality of the non-inferior solutions of MOGSA is higher than those of the other five algorithms under different parameter conditions.

Taking the parameter conditions of $\alpha = 0.9$, $CT_1 = 68$, and $CT_2 = 88$ as an example, two schemes S_1 and S_2 for obtaining the minimum smoothness index and the maximum disassembly profits on the parallel disassembly lines are given, as shown in Table 18. In scheme S_1, the minimum number of workers is 7, the minimum smoothness index is 9.91, and the profits are 929.88. In scheme S_2, the maximum disassembly profits are 1183.1, the number of stations is 9, and the smoothness index is 24.51. The two disassembly schemes have a different emphasis on different objectives, and it is impossible to optimize all the objectives at the same time. The numbers of tasks for televisions and refrigerators are 27 and 25, respectively, while the numbers of tasks in scheme S_1 are 16 and 19, respectively, indicating that partial disassembly can reduce the number of tasks. The numbers of tasks for televisions and refrigerators in scheme S_2 are 26 and 25, respectively, which are similar to complete disassembly. These results show that the more tasks that are disassembled, the greater the profits that are obtained, which is consistent with the theory. Additionally, the utilization rate of workstations in both schemes is very high. The utilization rate of seven workstations in scheme S_1 is over 99.52%, that of seven workstations in scheme S_2 is over 98.94%, and the maximum utilization rate of both schemes is 99.98%.

5.4. Discussion

The stochastic parallel assembly line balancing in Section 5.1 has many similarities with the stochastic parallel partial disassembly line balancing in this paper, such as the stochastic operating time, parallel layout, and precedence constraints. However, the model in this paper is more complex than that in the literature because there are more objective space and more feasible solutions in partial operations. Complete line balancing is only a special case of partial line balancing. The proposed GSA can obtain 101 better solutions among 372 problems, which shows that the performance of the proposed algorithm is better than TS in the literature, and the proposed algorithm can also be applied to the same types of assembly line balancing problems.

Section 5.2 verifies that the performance of the proposed algorithm is better than that of the exact method. For small-scale problems, the exact method can obtain the optimal solution while the meta-heuristic method cannot determine whether the solution is optimal, so the two methods can mutually verify each other. The advantages of the meta-heuristic method become more obvious when the scale of the problem increases. Besides, the performance of the hybrid algorithm is improved over the performance of the basic algorithms. Therefore, the hybrid algorithm is an important way to improve the performance of the method and optimize the complex combinatorial optimization problem.

The practical application of the proposed model and algorithm is realized in Section 5.3. The proposed multi-objective algorithm is superior to two multi-objective algorithms with local search strategies and three general multi-objective algorithms. The reason is that the hybrid algorithm considers both the global search of the genetic algorithm and the local search of simulated annealing. The utilization rate of the workstation in the obtained disassembly schemes is very high. There are two reasons: first, the proposed model adopts an adaptive cycle time, which can reduce the idle time of stations; and second, the proposed algorithm has better optimization performance, which can reasonably select and assign tasks to each station. Decision-makers can focus on one or two objectives when choosing disassembly schemes. The non-inferior solutions in Fig. 10 are high-quality disassembly schemes, which expand the decision-making space for decision-makers.

6. Conclusions and future work

To efficiently disassemble multi-variety and large-scale waste electrical and electronic equipment, this paper establishes a stochastic parallel partial disassembly line balancing model that uses the number of stations, smoothness index, and disassembly profits as evaluation indicators of disassembly lines. Then, a hybrid genetic algorithm and simulated annealing algorithm are proposed. The algorithm's encoding and decoding strategies consider both the disassembly task sequence and the disassembly task variable. The operations of the algorithm based on the precedence constraints can ensure that the solutions are feasible and improve the search efficiency. Among the 372 stochastic parallel assembly line balancing problems, the proposed algorithm improves the solutions of 101 problems. Among 45 parallel disassembly line balancing problems, the proposed GSA is better than CPLEX and basic GA and SA. Moreover, the stability and convergence of the proposed MOGSA are better than those of the MOSA, HPSOSA, NSGA-II, PESA-II, and SPEA2 in the stochastic parallel partial disassembly lines consisting of waste televisions and refrigerators. The proposed algorithm can obtain multiple preferred disassembly schemes. The results show that the proposed method can improve the performance of disassembly lines.

In future research, the proposed algorithm can be applied to other combinatorial optimization problems, such as newsvendor problems [61], vehicle routing problems, and job shop scheduling problems. In addition, various strategies are adopted to improve the convergence performance of the algorithm [62,63]. Additionally, it is possible to increase the evaluation indicators of the disassembly line, such as disassembly energy consumption and disassembly complexity.

CRediT authorship contribution statement

Kaipu Wang: Conceptualization, Methodology, Software, Validation, Investigation, Writing - original draft, Visualization. **Xinyu Li:** Methodology, Formal analysis, Data curation, Supervision, Writing - review & editing. **Liang Gao:** Conceptualization, Funding acquisition, Resources, Project administration, Writing - review & editing. **Peigen Li:** Methodology, Formal analysis, Supervision, Writing - review & editing. **Surendra M. Gupta:** Methodology, Formal analysis, Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Key Research and Development Project of China (Grant No. 2019YFB1704600), the National Natural Science Foundation of China (Grant Nos. 51825502 and 51775216), the Program for HUST Academic Frontier Youth Team (Grant No. 2017QYTD04), and the China Scholarship Council (Grant No. 202006160108).

Appendix

See Table A.1, Table A.2, Fig. A.1, and Fig. A.2.

Table A.1
Disassembly information about waste television.

Task no.	Description	Mean	Deviation	Revenue	Hazard
A1	Type A screw	18.4	4.6	0.55	0
A2	Back cover	17.4	4.4	0.37	0
A3	Antenna	5.4	1.4	0.11	0
A4	Horn	12.8	3.2	1.59	0
A5	Wire	17.0	4.3	0.28	0
A6	Main circuit board	6.0	1.5	1.83	1
A7	Power cord	18.2	4.6	0.10	0
A8	Deflecting coil	17.2	4.3	1.28	0
A9	Adjustment ring	6.8	1.7	0.94	0
A10	High pressure cap	9.2	2.3	2.52	1
A11	Degaussing coil	6.8	1.7	0.80	0
A12	Ground wire	7.6	1.9	0.13	0
A13	Transformer	8.6	2.2	0.93	0
A14	Low noise block	9.2	2.3	3.38	1
A15	Circuit board	10.4	2.6	4.27	1
A16	Glass tube	8.6	2.2	1.20	0
A17	Electron gun	8.0	2.0	3.30	1
A18	Explosion proof belt	9.2	2.3	0.81	0
A19	Type B screw	10.2	2.6	0.34	0
A20	Front shell	9.6	2.4	0.16	0
A21	Cathode ray tube	13.4	3.4	2.63	1
A22	Conical glass	6.6	1.7	0.66	1
A23	Phosphor	7.8	2.0	0.41	0
A24	Threaded peg	2.8	0.7	0.57	0
A25	Anode cap	4.6	1.2	0.93	0
A26	Cathode cover	5.2	1.3	0.66	0
A27	Screen glass	5.4	1.4	0.30	0

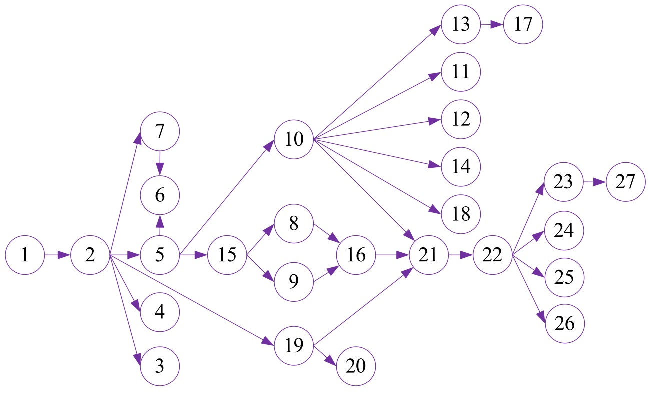


Fig. A.1. Precedence relationships of parts of the waste television.

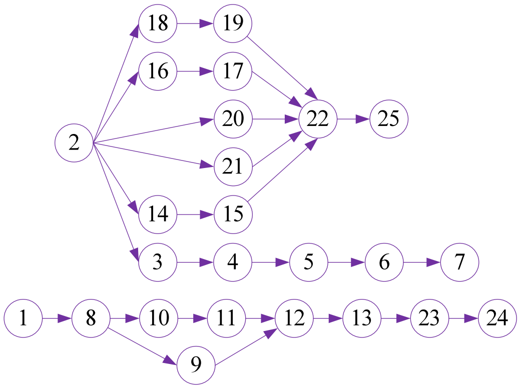


Fig. A.2. Precedence relationships of parts of the waste refrigerator.

Table A.2
Disassembly information about the waste refrigerator.

Task no.	Description	Mean	Deviation	Revenue	Hazard
B1	Compressor cover plate	18.6	4.7	1.55	0
B2	Door shell	15.1	3.8	0.95	0
B3	Plastic box	4.4	1.1	0.22	0
B4	Insulating glass	5.5	1.4	0.55	0
B5	Sealing ring	7.6	1.9	1.32	0
B6	Door liner	11.6	2.9	3.25	0
B7	Insulation layer	9.5	2.4	3.35	0
B8	Recycling refrigerant	40.8	10.2	4.55	1
B9	Compressor seat	15.7	3.9	2.35	0
B10	Wire 1	6.5	1.6	0.11	0
B11	Circuit board	8.6	2.1	4.27	1
B12	Radiator	6.5	1.6	1.59	0
B13	Compressor	16.2	4.1	2.52	1
B14	Fan	15.4	3.8	0.41	0
B15	Timer	16.0	4.0	0.93	0
B16	Control box	9.6	2.4	1.59	1
B17	Floodlight	13.4	3.4	0.75	0
B18	Wire 2	3.3	0.8	0.28	0
B19	Main circuit board	7.6	1.9	2.63	1
B20	Cabinet interior shelf	4.4	1.1	0.81	0
B21	Plastic box	4.4	1.1	0.13	0
B22	Condenser	13.5	3.4	0.66	1
B23	Compressor punch	25.0	6.2	1.25	0
B24	Recovery compressor oil	30.3	7.6	0.66	1
B25	Crushing box	20.4	5.1	0.57	0

References

[1] K. Wang, X. Li, L. Gao, A. Garg, Partial disassembly line balancing for energy consumption and profit under uncertainty, Robot. Comput.-Integr. Manuf. 59 (2019) 235–251.

[2] Y. Kazancoglu, Y. Ozturkoglu, Integrated framework of disassembly line balancing with green and business objectives using a mixed MCDM, J. Cleaner Prod. 191 (2018) 179–191.

[3] A. Gungor, S.M. Gupta, Disassembly line in product recovery, Int. J. Prod. Res. 40 (11) (2002) 2569–2589.

[4] L. Ozbakir, A. Baykasoglu, B. Gorkemli, L. Gorkemli, Multiple-colony ant algorithm for parallel assembly line balancing problem, Appl. Soft Comput. 11 (3) (2011) 3186–3198.

[5] S. Hezer, Y. Kara, A network-based shortest route model for parallel disassembly line balancing problem, Int. J. Prod. Res. 53 (6) (2015) 1849–1865.

[6] M.L. Bentaha, O. Battaia, A. Dolgui, S.J. Hu, Second order conic approximation for disassembly line design with joint probabilistic constraints, European J. Oper. Res. 247 (3) (2015) 957–967.

[7] M.H. Alavidoost, M. Tarimoradi, M.H.F. Zarandi, Fuzzy adaptive genetic algorithm for multi-objective assembly line balancing problems, Appl. Soft Comput. 34 (2015) 655–677.

[8] F.T. Altekin, L. Kandiller, N.E. Ozdemirel, Profit-oriented disassembly-line balancing, Int. J. Prod. Res. 46 (10) (2008) 2675–2693.

[9] K. Wang, X. Li, L. Gao, Modeling and optimization of multi-objective partial disassembly line balancing problem considering hazard and profit, J. Cleaner Prod. 211 (2019) 115–133.

[10] N. Deniz, F. Ozcelik, An extended review on disassembly line balancing with bibliometric & social network and future study realization analysis, J. Cleaner Prod. 225 (2019) 697–715.

[11] S. Agrawal, M.K. Tiwari, A collaborative ant colony algorithm to stochastic mixed-model U-shaped disassembly line balancing and sequencing problem, Int. J. Prod. Res. 46 (6) (2008) 1405–1429.

[12] K. Wang, X. Li, L. Gao, A multi-objective discrete flower pollination algorithm for stochastic two-sided partial disassembly line balancing problem, Comput. Ind. Eng. 130 (2019) 634–649.

[13] K. Wang, L. Gao, X. Li, A multi-objective algorithm for U-shaped disassembly line balancing with partial destructive mode, Neural Comput. Appl. 32 (16) (2020) 12715–12736.

[14] M.L. Bentaha, A. Dolgui, O. Battaia, R.J. Riggs, J. Hu, Profit-oriented partial disassembly line design: dealing with hazardous parts and task processing times uncertainty, Int. J. Prod. Res. 56 (24) (2018) 7220–7242.

[15] M.L. Bentaha, O. Battaia, A. Dolgui, A sample average approximation method for disassembly line balancing problem under uncertainty, Comput. Oper. Res. 51 (2014) 111–122.

[16] E. Tuncel, A. Zeid, S. Kamarthi, Solving large scale disassembly line balancing problem with uncertainty using reinforcement learning, J. Intell. Manuf. 25 (4) (2014) 647–659.

[17] F.T. Altekin, Z.P. Bayindir, V. Gumuskaya, Remedial actions for disassembly lines with stochastic task times, Comput. Ind. Eng. 99 (2016) 78–96.

[18] F.T. Altekin, A comparison of piecewise linear programming formulations for stochastic disassembly line balancing, Int. J. Prod. Res. 55 (24) (2017) 7412–7434.

[19] F. Zheng, J. He, F. Chu, M. Liu, A new distribution-free model for disassembly line balancing problem with stochastic task processing times, Int. J. Prod. Res. 56 (24) (2018) 7341–7353.

[20] C.B. Kalayci, A. Hancilar, A. Gungor, S.M. Gupta, Multi-objective fuzzy disassembly line balancing using a hybrid discrete artificial bee colony algorithm, J. Manuf. Syst. 37 (2015) 672–682.

[21] Z. Zhang, K. Wang, L. Zhu, Y. Wang, A Pareto improved artificial fish swarm algorithm for solving a multi-objective fuzzy disassembly line balancing problem, Expert Syst. Appl. 86 (2017) 165–176.

[22] S.M. McGovern, S.M. Gupta, A balancing method and genetic algorithm for disassembly line balancing, European J. Oper. Res. 179 (3) (2007) 692–708.

[23] S.M. McGovern, S.M. Gupta, Ant colony optimization for disassembly sequencing with multiple objectives, Int. J. Adv. Manuf. Technol. 30 (5–6) (2006) 481–496.

[24] K. Wang, X. Li, L. Gao, P. Li, Energy consumption and profit-oriented disassembly line balancing for waste electrical and electronic equipment, J. Cleaner Prod. 265 (2020) 121829.

[25] Y. Ren, D. Yu, C. Zhang, G. Tian, L. Meng, X. Zhou, An improved gravitational search algorithm for profit-oriented partial disassembly line balancing problem, Int. J. Prod. Res. 55 (24) (2017) 7302–7316.

[26] F. Pistolesi, B. Lazzerini, M.D. Mura, G. Dini, EMOGA: A hybrid genetic algorithm with extremal optimization core for multiobjective disassembly line balancing, IEEE Trans. Ind. Inf. 14 (3) (2018) 1089–1098.

[27] K. Wang, X. Li, L. Gao, P. Li, Modeling and balancing for green disassembly line using associated parts precedence graph and multi-objective genetic simulated annealing, Int. J. Precis. Eng. Manuf.-Green Technol. (2020) http://dx.doi.org/10.1007/s40684-020-00259-7.

[28] E. Özceylan, C.B. Kalayci, A. Güngör, S.M. Gupta, Disassembly line balancing problem: a review of the state of the art and future directions, Int. J. Prod. Res. 57 (15–16) (2019) 4805–4827.

[29] S.M. McGovern, S.M. Gupta, Combinatorial optimization analysis of the unary NP-complete disassembly line balancing problem, Int. J. Prod. Res. 45 (18–19) (2007) 4485–4511.

[30] M.L. Bentaha, O. Battaia, A. Dolgui, An exact solution approach for disassembly line balancing problem under uncertainty of the task processing times, Int. J. Prod. Res. 53 (6) (2015) 1807–1818.

[31] Y. Ren, C. Zhang, F. Zhao, G. Tian, W. Lin, L. Meng, H. Li, Disassembly line balancing problem using interdependent weights-based multi-criteria decision making and 2-optimal algorithm, J. Cleaner Prod. 174 (2018) 1475–1486.

[32] R.J. Riggs, O. Battaia, S.J. Hu, Disassembly line balancing under high variety of end of life states using a joint precedence graph approach, J. Manuf. Syst. 37 (2015) 638–648.

[33] S. Avikal, R. Jain, P.K. Mishra, A Kano model, AHP and M-TOPSIS method-based technique for disassembly line balancing under fuzzy environment, Appl. Soft Comput. 25 (2014) 519–529.

[34] S. Avikal, P.K. Mishra, R. Jain, A fuzzy AHP and PROMETHEE method-based heuristic for disassembly line balancing problems, Int. J. Prod. Res. 52 (5) (2014) 1306–1317.

[35] C.B. Kalayci, O. Polat, S.M. Gupta, A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem, Ann. Oper. Res. 242 (2) (2016) 321–354.

[36] C.B. Kalayci, S.M. Gupta, A particle swarm optimization algorithm with neighborhood-based mutation for sequence-dependent disassembly line balancing problem, Int. J. Adv. Manuf. Technol. 69 (1–4) (2013) 197–209.

[37] C.B. Kalayci, S.M. Gupta, Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem, Expert Syst. Appl. 40 (18) (2013) 7231–7241.

[38] C.B. Kalayci, S.M. Gupta, A tabu search algorithm for balancing a sequence-dependent disassembly line, Prod. Plan. Control 25 (2) (2014) 149–160.

[39] C.B. Kalayci, O. Polat, S.M. Gupta, A variable neighbourhood search algorithm for disassembly lines, J. Manuf. Technol. Manag. 26 (2) (2015) 182–194.

[40] U. Özcan, Balancing stochastic parallel assembly lines, Comput. Oper. Res. 99 (2018) 109–122.

[41] Y. Fang, H. Ming, M. Li, Q. Liu, P. Duc Truong, Multi-objective evolutionary simulated annealing optimisation for mixed-model multi-robotic disassembly line balancing with interval processing time, Int. J. Prod. Res. 58 (3) (2020) 846–862.

[42] A. Aydemir-Karadag, O. Turkbey, Multi-objective optimization of stochastic disassembly line balancing with station paralleling, Comput. Ind. Eng. 65 (3) (2013) 413–425.

[43] L.-P. Ding, Y.-X. Feng, J.-R. Tan, Y.-C. Gao, A new multi-objective ant colony algorithm for solving the disassembly line balancing problem, Int. J. Adv. Manuf. Technol. 48 (5–8) (2010) 761–771.

[44] Y. Fang, Q. Liu, M. Li, Y. Laili, P. Duc Truong, Evolutionary many-objective optimization for mixed-model disassembly line balancing with multi-robotic workstations, European J. Oper. Res. 276 (1) (2019) 160–174.

[45] J. Liu, Z. Zhou, P. Duc Truong, W. Xu, J. Yan, A. Liu, C. Ji, Q. Liu, An improved multi-objective discrete bees algorithm for robotic disassembly line balancing problem in remanufacturing, Int. J. Adv. Manuf. Technol. 97 (9–12) (2018) 3937–3962.

[46] L. Zhu, Z. Zhang, Y. Wang, A Pareto firefly algorithm for multi-objective disassembly line balancing problems with hazard evaluation, Int. J. Prod. Res. 56 (24) (2018) 7354–7374.

[47] A. Gungor, S.M. Gupta, A solution approach to the disassembly line balancing problem in the presence of task failures, Int. J. Prod. Res. 39 (7) (2001) 1427–1467.

[48] E.G. Kalaycilar, M. Azizoglu, S. Yeralan, A disassembly line balancing problem with fixed number of workstations, European J. Oper. Res. 249 (2) (2016) 592–604.

[49] S. Mete, Z.A. Cil, K. Agpak, E. Ozceylan, A. Dolgui, A solution approach based on beam search algorithm for disassembly line balancing problem, J. Manuf. Syst. 41 (2016) 188–200.

[50] J. Li, X. Chen, Z. Zhu, C. Yang, C. Chu, A branch, bound, and remember algorithm for the simple disassembly line balancing problem, Comput. Oper. Res. 105 (2019) 47–57.

[51] S. Mete, Z.A. Cil, E. Celik, E. Ozceylan, Supply-driven rebalancing of disassembly lines: A novel mathematical model approach, J. Cleaner Prod. 213 (2019) 1157–1164.

[52] Z.A. Çil, S. Mete, E. Özceylan, K. Ağpak, A beam search approach for solving type II robotic parallel assembly line balancing problem, Appl. Soft Comput. 61 (2017) 129–138.

[53] H. Gökçen, K. Ağpak, R. Benzer, Balancing of parallel assembly lines, Int. J. Prod. Econ. 103 (2) (2006) 600–609.

[54] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[55] S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb, A simulated annealing-based multiobjective optimization algorithm: AMOSA, IEEE Trans. Evol. Comput. 12 (3) (2008) 269–283.

[56] H. Tang, R. Chen, Y. Li, Z. Peng, S. Guo, Y. Du, Flexible job-shop scheduling with tolerated time interval and limited starting time interval based on hybrid discrete PSO-SA: An application from a casting workshop, Appl. Soft Comput. 78 (2019) 176–194.

[57] D.W. Corne, N.R. Jerram, J.D. Knowles, M.J. Oates, PESA-II: Region-based selection in evolutionary multiobjective optimization, in: L. Spector, E.D. Goodman, A. Wu, W.B. Langdon, H.-M. Voigt (Eds.) Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, 2001, pp. 283-290.

[58] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, in: K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou, T. Fogarty (Eds.) Evolutionary Methods for Design, Optimisation and Control, 2002, pp. 95-100.

[59] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Trans. Evol. Comput. 3 (4) (1999) 257–271.

[60] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G.d. Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Trans. Evol. Comput. 7 (2) (2003) 117–132.

[61] F. Wang, Y. Li, A. Zhou, K. Tang, An estimation of distribution algorithm for mixed-variable newsvendor problems, IEEE Trans. Evol. Comput. 24 (3) (2020) 479–493.

[62] F. Wang, H. Zhang, A. Zhou, A particle swarm optimization algorithm for mixed-variable optimization problems, Swarm Evol. Comput. 60 (2021) 100808.

[63] F. Wang, Y. Li, F. Liao, H. Yan, An ensemble learning based prediction strategy for dynamic multi-objective optimization, Appl. Soft Comput. 96 (2020) 106592.