sustainability

MDPI

Article

Energy-Efficient Optimization of Two-Sided Disassembly Line Balance Considering Parallel Operation and Uncertain Using Multiobjective Flatworm Algorithm

Junyong Liang [1,2,*], Shunsheng Guo [1,*], Yunfei Zhang [2], Wenfang Liu [2] and Shengwen Zhou [1]

[1] School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China; swzhou1007@163.com
[2] School of Civil Engineering, Sichuan University of Science and Engineering, Zigong 643000, China; yf_zhang@suse.edu.cn (Y.Z.); wf_liu@suse.edu.cn (W.L.)
* Correspondence: liangjunyong@suse.edu.cn (J.L.); guoshunsheng@whut.edu.cn (S.G.)

Abstract: The two-sided disassembly line is popular for its high-efficiency disassembly of large-volume end-of-life products. However, in the process of two-sided disassembly, some parts and components need to be disassembled in parallel, and the uncertainty of disassembly time lacks certain research. This paper constructs a fuzzy multiobjective two-sided disassembly line balance problem model based on parallel operation constraint, which aims to reduce the balance loss rate, smoothness index, and energy consumption of disassembly activities. A multiobjective flatworm algorithm based on the Pareto-dominance relationship is developed. To increase the diversity of feasible solutions in the evolution process and accelerate the convergence of Pareto-optimal solutions to prevent the random search of solution space, growth, splitting and regeneration mechanisms are embedded in the algorithm. The working mechanism and efficiency of the multiobjective flatworm algorithm are proved on a series of two-sided disassembly cases, and the excellent performance of the proposed model and algorithm are demonstrated by an actual automobile two-sided disassembly line.

Keywords: energy-efficient optimization; two-sided disassembly line balance; parallel operation; uncertain; multiobjective flatworm algorithm

Citation: Liang, J.; Guo, S.; Zhang, Y.; Liu, W.; Zhou, S. Energy-Efficient Optimization of Two-Sided Disassembly Line Balance Considering Parallel Operation and Uncertain Using Multiobjective Flatworm Algorithm. *Sustainability* **2021**, *13*, 3358. https://doi.org/10.3390/su13063358

Academic Editor: Maxim A. Dulebenets

Received: 5 March 2021
Accepted: 16 March 2021
Published: 18 March 2021

1. Introduction

With the advancement of manufacturing processes, the lifecycle of products is constantly shortened, and more end-of-life (EOL) products are becoming increasingly unavailable [1]. The "circular economy" and "remanufacturing" industries are in the ascendant, the recycling of end-of-life products such as black and white home appliances and automobiles has received increasing attention. The latest data showed that the number of EOL automobiles in China was about 2.615 million in 2020, the total weight of EOL vehicles was nearly 6.267 million tons, and each ton of the resources can recover 72.0% of waste steel, 6.0% of waste nonferrous metals, 6.3% of waste plastics, 4.4% of waste rubber and 1.7% of waste glass; if they are directly discarded, it will not only waste resources but also pollute the environment [2]. Disassembly, as an important stage after recycling and before reuse, plays an important role in the process from waste to available resources [3]. When dealing with a large number of product disassemblies, the paced disassembly line has a great advantage of efficiency compared with the fixed disassembly position layout [4]. How to assign disassembly tasks to reasonable stations in proper order under given constraints, and achieve the expected objectives, such as reducing the cycle time of disassembly line and improve the efficiency of comprehensive disassembly, is called the disassembly line balancing problem [5,6].

The traditional straight one-sided and U-shaped lines are mainly used to dismantle small-sized and medium-sized EOL products, such as TV sets, microwave ovens and other

white appliances [7], and tasks can only be carried out in the space on one side of the disassembly line. Facing a large number of large-volume EOL products to be disassembled, especially for the EOL automobiles and refrigerators, if these two disassembly lines are used, more space, more tools and resources will not only be occupied, but will also lead to a lower disassembly efficiency. Under this situation, the two-sided disassembly line is a better choice for these large-volume EOL products [8,9], tasks can be performed on the left and right sides of the two-sided disassembly line at the same time, which can reduce the length of the disassembly line to reduce plant space occupation; meanwhile, it can realize the shared use of tooling and fixtures to reduce the number of disassembly equipment and corresponding energy consumption.

The disassembly line balancing problem (DLBP) has received widespread attention since it was first proposed [10], and it is considered to be non-deterministic polynomial complete (NP-complete) [11]. There are roughly three categories of methods to solve the DLBP: exact methods, heuristic methods and metaheuristic algorithms. In the exact methods, mixed-integer linear programming [12–16], branch and bound [17,18] are mostly used to solve DLBP in the linear interactive and general optimizer (LINGO). Aiming at the low efficiency of solving large-scale DLBP problems, heuristic and metaheuristic algorithms are proposed. For heuristics, the greedy/2-opt algorithm is mainly applied [11]. For metaheuristics, such as genetic algorithm [19], ant colony optimization [20], particle swarm optimization [21], simulated annealing [22], artificial bee colony and variable neighborhood search [23]. In recent years, the Pareto firefly algorithm [8], flower pollination algorithm [9] and gravitational search algorithm [24] are also successfully applied in DLBP.

In terms of two-sided disassembly lines suitable for large-volume EOL products, few kinds of literature are related to relevant research. Kucukkoc described the two-sided disassembly line balancing problem (TDLBP) and proposed a mixed-integer linear programming model [25]. Wang proposed the stochastic two-sided partial disassembly line balancing problem and solved it by the discrete flower pollination algorithm [9].

Extensive literature shows that DLBP is not the reverse process of an assembly line [26]; disassembly operations are usually fraught with uncertainties. Three techniques are often used to deal with these uncertainties. Most literature assumes that the disassembly time is a random variable with a known probability distribution (such as the normal distribution) [9,13,27,28]. Some assume that the disassembly time is an interval number [22]. The disassembly time is expressed by triangular fuzzy numbers in other studies [19,29,30].

However, all of these methods dealing with the uncertain disassembly time have some limitations to some extent. In practice, it is often impossible to obtain the unbiased estimation of disassembly time by a limited number of samples, let alone the probability distribution function of disassembly time due to the high uncertainty of EOL product or the skill of the disassembly operator. The interval numbers of disassembly time only give the lower and upper bounds, which can be obtained easily but lack some necessary statistical significance. In addition, some practical constraints of large-volume EOL products in the two-sided disassembly line are idealized, which lead to the inconsistency between the previous two-sided disassembly line model and the reality. In view of the deficiencies of the above research, the main contributions of this paper follow:

1. The triangular fuzzy number is introduced into the two-sided disassembly line to simulate the uncertainty of parts disassembly time. This method can not only simulate the average disassembly time of parts, but also expresses the disassembly time of parts with good conditions (less than the average disassembly time) and parts with poor conditions (more than the average disassembly time), which increases the universality of uncertainty expression.
2. The parallel operation constraint is introduced to imitate some large-volume components and parts that lay across the two-side disassembly line and that should be dismantled parallel in the same mated-station, which is significantly different from the straight one-sided disassembly line and U-shaped. For example, the removal of the engine cover of an EOL automobile requires the workers at the left and right

stations of the same disassembly mated-station to work together to remove their respective fastening bolts due to its large size.

3. A mixed-integer programming (MIP) model based on energy efficiency is established for the two-sided disassembly line balancing problem considering parallel operation and fuzzy processing times (TDLBP-POF), which is more realistic than the previous model, and a multiobjective flatworm algorithm (MOFA) is developed to solve the problem.

The rest of the paper is organized as follows. Section 2 describes the problem description and constructs the mathematical model for TDLBP-POF. Then the proposed multiobjective flatworm algorithm for solving TDLBP-POF is explained in detail in Section 3. Section 4 presents three computational cases to verify the performance of the proposed model and algorithm. Conclusions, including directions for further research and exploration, are argued in the last section.

2. Model Construction of TDLBP-POF

2.1. Problem Description

The two-sided disassembly line is to arrange the workstations on the two sides of the conveyor to form two independent working areas. As shown in Figure 1, each station is equipped with operators, fixtures and disassembly tools; the EOL product is distributed to the left (L) or right (R) side of the disassembly line, and is broken down into the required parts on each workstation. Two workstations directly facing each other is called a mated-station, the station (2, L) and (2, R) together constitute the mated-station 2, station (2, L) can be called the companion station of the station (2, R), and vice versa [7,31].

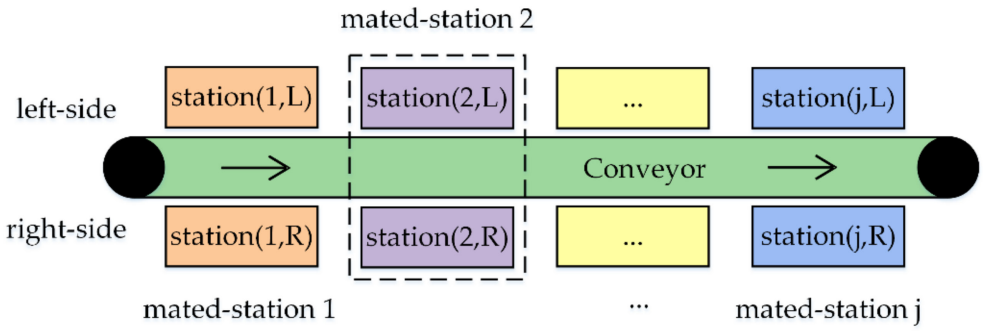

Figure 1. Station configuration of the two-sided disassembly line.

Figure 2 shows a two-sided disassembly problem with 16 tasks (P16) modified from the assembly line problem [32]. Each circle indicates a task, and the letters (L/R/E) above the circle represent the side of the task that can be arranged; tasks with L indicate that those tasks can only be dismantled on the left station of the disassembly line, and R on the right, tasks with E imply that they can be disassembled on the left as well as on the right of the line [7,9]. The real number between 0 and 1 in brackets above denotes the hazard degree of the task: the larger, the higher the danger degree is, and the more serious the environmental pollution generated. It should be noted that if the hazard degree of a task is 0, it is not identified in Figure 2. The triple array below the task represents the fuzzy disassembly time of the task, which will be introduced in Section 2.3.

The problem can be described as how to assign a sequence of disassembly tasks to workstations in a reasonable manner, so that the efficiency of the disassembly line is fully utilized and the task disassembly time on each workstation is approximately balanced; meanwhile, energy consumption during the disassembly process is effectively reduced.

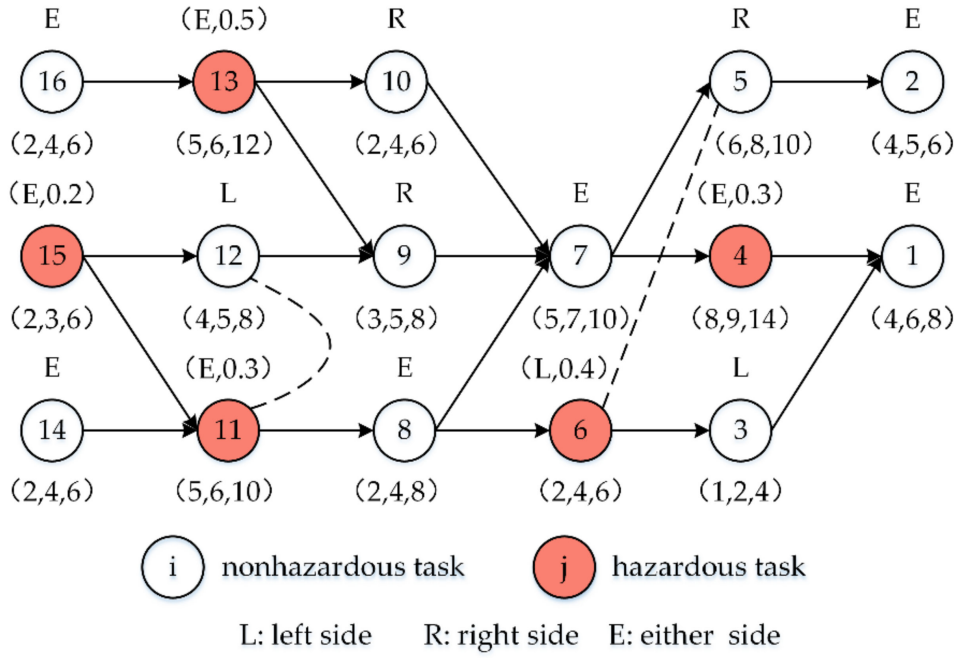

Figure 2. Disassembly precedence relationship with 16 tasks.

2.2. Parallel Operation Constraint

In the disassembly process of EOL products, in addition to meeting the basic task disassembly precedence constraints, due to the characteristics of large-volume EOL products, a parallel operation constraint often appears in actual disassembly, which means that a pair of tasks must be disassembled at the left and right stations of the same disassembly mated-station in parallel, whether the disassembly time of the pair of tasks is the same or not, and they usually have to be completed at the same time in order to remove the parts containing the pair of tasks. In Figure 2, a pair of tasks connected by a dotted line indicates that they need a parallel operation by two stations in a mated-station to complete the disassembly. For example, task 5 should be dismantled in a station on the right, task 6 is on the left in the same mated-station. It should be pointed out that the disassembly direction of a pair of tasks that need parallel operation must be compatible, that is, the disassembly direction of the pair of tasks cannot be L or R at the same time.

2.3. Approaches for Fuzzy Processing Times

To simulate the uncertainty or inaccuracy of disassembly time in a more realistic situation, triangular fuzzy numbers are applied to represent and describe the undetermined operation time due to their simplicity of calculation and are easy to compare with other fuzzy numbers [30]. Assuming that the disassembly time of task i is a triangular fuzzy number with three parameters $\widetilde{A} = \left(A^l, A^m, A^u\right)$, where A^l and A^u are the upper and lower bounds of the triangular fuzzy numbers, representing optimistic disassembly time and conservative disassembly time, respectively [29], the membership function value corresponding to A^m is equal to a maximum of 1, which indicates the most likely disassembly time in the process. Figure 3 describes a triangular fuzzy number of $\widetilde{A}$, the x-axis represents the boundary range of $\widetilde{A}$, and $\mu_{\widetilde{A}}(x)$ represents the membership degree of the fuzzy number $\widetilde{A}$, the membership degree decreases gradually from point A^m to both sides, and decreases to 0 at A^l and A^u.

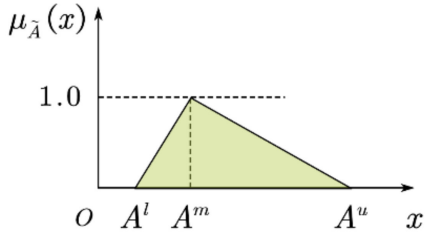

Figure 3. Membership function of a triangular fuzzy number.

Let $\widetilde{B} = \left(B^l, B^m, B^u\right)$ represent another triangular fuzzy number. When solving the TDLBP-POF, all of the time parameters are triangular fuzzy numbers; in order to calculate the start and end time of the disassembly tasks, basic operation rules of two triangular fuzzy numbers are calculated as follows:

$$\widetilde{A} + \widetilde{B} = \left(A^l + B^l, A^m + B^m, A^u + B^u\right) \tag{1}$$

$$\widetilde{A} - \widetilde{B} = \left((A^l - B^u) \vee 0, A^m - B^m, A^u + B^l\right) \tag{2}$$

$$\widetilde{A} \times \widetilde{B} = \left(A^l \times B^l, A^m \times B^m, A^u \times B^u\right) \tag{3}$$

$$\widetilde{A} / \widetilde{B} = \left(A^l / B^u, A^m / B^m, A^u / B^l\right) \tag{4}$$

The symbol $(A^l - B^u) \vee 0$ in Equation (2) represents the parameter at this position, taking the maximum value between $A^l - B^u$ and 0. Meanwhile, in order to compare the size of a group of fuzzy numbers, a popular defuzzification technique is adopted to convert fuzzy numbers into crisp real numbers [30,33], the defuzzification method can be computed by Equation (5). Notation $DF(\widetilde{A})$ indicates the maximum possible real number of $\widetilde{A}$.

$$DF(\widetilde{A}) = \left(A^l + 2A^m + A^u\right)/4 \tag{5}$$

For example, in Figure 2, the fuzzy disassembly time of task 8 is (2,4,8) and that of task 16 is (2,4,6). Although the most likely disassembly time in the middle is 4 units, the results are not the same when using Equation (5) to defuzzify them into real numbers because their upper and lower bounds are different, and be converted into 4.5 and 4.0 units, respectively.

2.4. Mathematical Model of TDLBP-POF

2.4.1. Notations

i, g, h : disassembly tasks index, $i, g, h \in I$.
j, p : mated-stations index, $j, p \in J$.
k : left or right side of a mated-station, $k = \begin{cases} 1, \, left\ side \\ 2, right\ side \end{cases}$.
(j, k) : the mated-station index j at side k.
N_t : number set of disassembly tasks.
N_m : number set of mated-stations.
I : disassembly task number, $I = \{1, 2, \cdots, N_t\}$.
J : mated-stations number, $J = \{1, 2, \cdots, N_m\}$.
$\widetilde{CT}$: the maximum allowed fuzzy cycle time.
$\widetilde{t}_i$: fuzzy time of duration of task i.
h_i : hazard degree of task i, $h_i \in [0, 1)$.
$\widetilde{T}_{jk}$: total fuzzy time of duration at the station (j, k).
I_l : set of tasks operated on the left of the disassembly line, $I_l \in I$.
I_r : set of tasks operated on the right of the disassembly line, $I_r \in I$.
I_e : set of tasks can be operated on the left or right side of the line, $I_e \in I$.
$P(i)$: set of immediate disassembly predecessors of task i.

$P_{all}(i)$: set of all disassembly predecessors of task i.
$S(i)$: set of immediate disassembly successors of task i.
$S_{all}(i)$: set of all disassembly successors of task i.
P_0 : set of tasks without immediate predecessors $P_0 = \{i \in I | P(i) = \varnothing\}$.
e_{ft} : unit energy consumption of fixtures and tools in each station.
e_{eq} : unit energy consumption of disassembly equipment in each station.
e_h : unit energy consumption of a hazardous task.
$O(i)$: set of tasks disassembled on the opposite side of task i.
$K(i)$: side of task i to be arranged, $K(i) = \begin{cases} 1, if\ i \in I_l \\ 2, if\ i \in I_r \\ \{1,2\}, if\ i \in I_e \end{cases}$.
TPO : set of disassembly tasks require a parallel operation.
$\widetilde{t}_i^s$: the fuzzy disassembly starting time of task i.
x_{ijk} : if task i is assigned to station (j,k), then $x_{ijk} = 1$; otherwise, $x_{ijk} = 0$.
$N^1(j)$: if only one station in the mated-station j is opened, $N^1(j) = 1$; otherwise, $N^1(j) = 0$.
$N^2(j)$: if a pair of stations in the mated-station j are opened, $N^2(j) = 1$; otherwise, $N^2(j) = 0$.
ξ : an infinite positive number.
η : energy consumption sharing coefficient for both a pair of stations are utilized in the same mated-station.
r_i : position index of task i in the disassembly sequence.
y_{ig} : if task i starts earlier than task g in the same station, then $y_{ig} = 1$; otherwise, $y_{ig} = 0$.
$\widetilde{BLR}$: the fuzzy line balance loss rate.
$\widetilde{SI}$: the fuzzy smoothness index.
$\widetilde{TEC}$: the fuzzy total energy consumption.

2.4.2. Objective Functions

$$\min \widetilde{BLR} = 1 - \frac{\sum_{i \in I} \widetilde{t}_i}{\widetilde{CT} \times \sum_{j \in J}(N^1(j) + 2 \times N^2(j))} \tag{6}$$

$$\min \widetilde{SI} = \sum_{j \in J} \sum_{k=1,2} (\widetilde{T}_{jk} - max\left\{\widetilde{T}_{jk}\right\}_{j \in J, k=1,2})^2 \tag{7}$$

$$\min \widetilde{TEC} = \widetilde{EC_1} + \widetilde{EC_2} + \widetilde{EC_3} \tag{8}$$

$$\widetilde{EC_1} = e_{ft} \times [2 \times \eta \times \sum_{j \in J} N^2(j) + \sum_{j \in J} N^1(j)] \times \widetilde{CT} \tag{9}$$

$$\widetilde{EC_2} = e_{eq} \times \sum_{i \in I} \widetilde{t}_i \tag{10}$$

$$\widetilde{EC_3} = e_h \times \sum_{i \in I} (1 + \frac{r_i}{N_t}) \times h_i \times \widetilde{t}_i \tag{11}$$

Three objectives need to be optimized in the TDLBP-POF. The fuzzy balance loss rate is expressed by Equation (6) to calculate the imbalance of the disassembly line, the lower $\widetilde{BLR}$ and the higher balance rate of the disassembly line; therefore, the minimal $\widetilde{BLR}$ is preferred in this paper. The smoothness index is designed to measure the workload balance or deviation of the workload among workstations; it is computed by Equation (7). The larger $\widetilde{SI}$ means the greater gap of working state and idle state between different workstations [34,35], so the minimal $\widetilde{SI}$ is pursued and zero is the ideal. The fuzzy total energy consumption is calculated by Equation (8), which consists of three parts, $\widetilde{EC_1}$ in Equation (9) represents the energy consumption caused by the lighting and ventilation of

the utilized stations, $\widetilde{EC_2}$ indicates the energy consumption related to disassembly time, and $\widetilde{EC_3}$ denotes the energy consumption of the hazardous task, which is not only related to the hazard degree of tasks, but also related to the disassembly sequence of the hazard tasks—tasks with higher hazard degree and later disassembly, which will consume more energy. Therefore, total energy consumption is the main focus of this paper, and the minimum is preferred.

2.4.3. Constrains

$$\sum_{j=1}^{N_m} \sum_{k=1,2} x_{ijk} = 1, \forall i \in I \tag{12}$$

$$\sum_{j=1}^{N_m} \sum_{k \in K(g)} j \cdot x_{gjk} \leq \sum_{j=1}^{N_m} \sum_{k \in K(h)} p \cdot x_{hpk}, \forall h \in I, g \in P(h) \tag{13}$$

$$\widetilde{t}_i^s + \widetilde{t}_i \leq \widetilde{CT}, \forall i \in I \tag{14}$$

$$\widetilde{t}_h^s - \left(\widetilde{t}_g^s + \widetilde{t}_g\right) + \xi \cdot \left(1 - \textstyle\sum_{k \in K(h)} x_{hjk}\right) + \xi \cdot \left(1 - \textstyle\sum_{k \in K(h)} x_{gjk}\right) \geq 0, \quad \forall h \in I - P_0, g \in P(h), j \in J \tag{15}$$

$$\widetilde{t}_g^s - (\widetilde{t}_i^s + \widetilde{t}_i) + \xi \cdot \left(1 - x_{ijk}\right) + \xi \cdot \left(1 - x_{gjk}\right) + \xi \cdot (1 - y_{ig}) \geq 0, \quad \forall i \in I, g \in \{m | m \in I - P_{all}(i) \cup S_{all}(i) \cup O(i) \text{ and } i < m\}, j \in J, k \in K(i) \cap K(g) \tag{16}$$

$$\widetilde{t}_i^s - (\widetilde{t}_g^s + \widetilde{t}_g) + \xi \cdot \left(1 - x_{ijk}\right) + \xi \cdot \left(1 - x_{gjk}\right) + \xi \cdot y_{ig} \geq 0, \quad \forall i \in I, g \in \{m | m \in I - P_{all}(i) \cup S_{all}(i) \cup O(i) \text{ and } i < m\}, j \in J, k \in K(i) \cap K(g) \tag{17}$$

$$\sum_{k_1 \in K(g)} x_{gjk_1} - \sum_{k_2 \in K(h)} x_{hjk_2} = 0, \forall (g,h) \in TCO, j \in J, k_1 \neq k_2 \tag{18}$$

$$\widetilde{t}_g^s + \widetilde{t}_g = \widetilde{t}_h^s + \widetilde{t}_h, \forall (g,h) \in TCO \tag{19}$$

$$x_{ij1} = 0, \forall i \in I_r, j \in J \tag{20}$$

$$x_{ij2} = 0, \forall i \in I_l, j \in J \tag{21}$$

$$\widetilde{t}_i^s \geq 0, \forall i \in I \tag{22}$$

Equation (12) indicates that each disassembly task can only be assigned to only one workstation. Equation (13) denotes that the disassembly precedence relations between two tasks should not be violated when disassembled at each workstation. For task pairs (g,h) where $g \in P(h)$, task h only can be disassembled after the predecessor task g is completed. Equation (14) ensures that the total disassembly time of any workstation is less than or equal to the allowed fuzzy cycle time $\widetilde{CT}$. Equations (15)–(17) ensure there is no time crossing between the start and end times of any two tasks with disassembly precedence relations at the same station, that is, an operator of any workstation can only execute at most one task at any time. An infinite positive number ξ is introduced to call off the constraints when the two tasks are scheduled to different mated-stations [7,36]. Equations (18) and (19) ensure that the completion time of two tasks that need to be operated in parallel at a mated-station is the same. Equations (20) and (21) are the disassembly direction constraints and the last ensures that all tasks start at a meaningful time.

3. Multiobjective Flatworm Algorithm for TDLBP-POF

The flatworm algorithm (FA) was recently developed by Tseng [37] to solve the problem of disassembly sequence planning, which proved that FA is more effective than either the genetic algorithm (GA) and ant colony algorithm (ACO) in reducing the times of disassembly direction change and tool replacement through a ceiling fan disassembly with 26 parts, a printer disassembly with 52 parts and a simulated product disassembly

with 150 parts. The main reason is that FA uses a single individual to generate the next generation, rather than cross-exchanging information between multiple individuals, which can effectively escape the local optimum and find a better solution. The algorithm was inspired by the self-renewal function of the multipotential stem cells (MSCs) of flatworms. A flatworm is accidentally split into two or more segments in the face of stimulation such as auto division or any disruption. Each segment will supplement the missing part of its full body through the growth and tissue repair function of its preserved multipotential stem cells. After a period of recovery, each part body of the flatworm will differentiate into a complete flatworm. FA is significantly different from the traditional GA, particle swarm optimization (PSO) and ACO. The population of FA constantly evolves through growth, splitting and regeneration mechanisms, with few evolutionary parameters and high efficiency, and only the population number of flatworms (n), growth probability (p) and the total evolutionary iterations are considered as the parameters.

This paper focuses on the multiobjective energy-efficient optimization of the two-sided disassembly line balancing, so the Pareto-optimal mechanism [38] is embedded into FA and called the multiobjective flatworm algorithm (MOFA); the other details of the growth, splitting and regeneration process are introduced in Sections 3.1–3.5. The procedure of the MOFA is depicted in Figure 4.

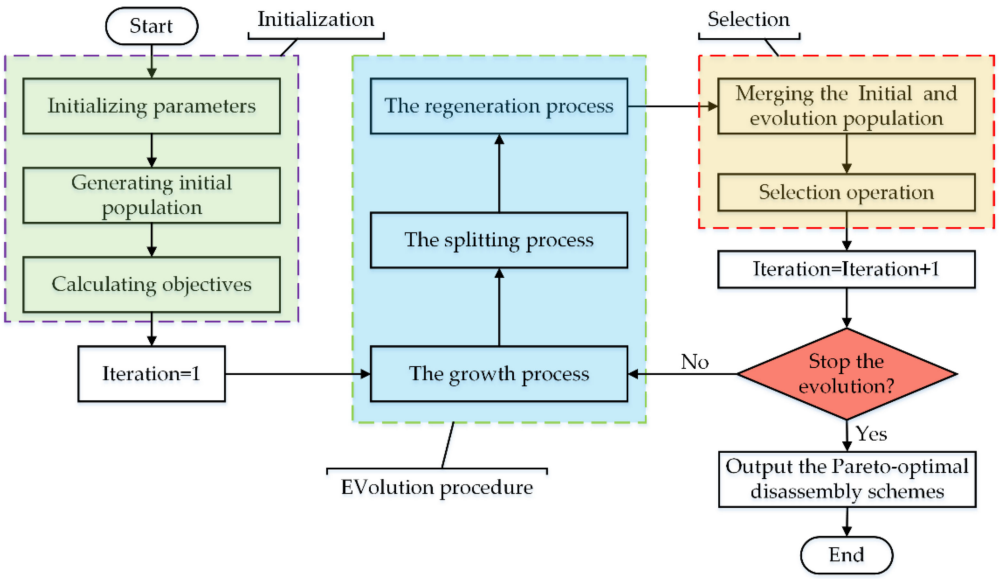

Figure 4. The procedure of the proposed multiobjective flatworm algorithm (MOFA).

3.1. Solution Encoding

A reasonable solution encoding scheme is a prerequisite for solving the TDLBP-POF problem with MOFA, so it should be a feasible disassembly task sequence, and meanwhile, conducive to the stable execution of growth, splitting and regeneration processes in the discrete solution space. Thus, an initial feasible solution is represented by a complete flatworm, in which a total of three vectors are employed in FA: the task disassembly sequence vector, the task disassembly direction vector and the task hazard degree vector. The first vector is composed of all tasks to determine the sequence to enter the disassembly line. The second represents the disassembly direction of each task, tasks with L indicate that those tasks can only be dismantled on the left station of the disassembly line, and R on the right, tasks with E imply that they can be disassembled on the left as well as on the right of the line. The hazard degree vector is used to indicate the environmental hazard degrees of disassembly tasks. It should be pointed out that, between a pair of tasks with parallel operation constraint, there should be no predecessor tasks of the pair tasks that finally appear in the disassembly task sequence. Figure 5 depicts a feasible encoding

scheme of P16. For example, task 13 is in the fifth position from the left of the disassembly sequence; its disassembly direction is E, and its hazard degree is 0.5.

task sequence	15	14	11	16	13	10	8	9	7	6	12	3	4	5	1	2
task disassembly direction	E	E	E	E	E	R	E	R	E	L	L	L	E	R	E	E
task hazard degree	0.2	0	0.3	0	0.5	0	0	0	0	0.4	0	0	0.3	0	0	0

Figure 5. A feasible encoding scheme of 16 disassembly tasks.

3.2. The Growth Process

As time goes on, the MSCs of a flatworm self-proliferate a limited number of times, and its body gradually grows longer, which makes the tissue preparation for the splitting process in the face of adverse stimulation in the future. In the problem of TDLBP-POF, the algorithm imitates the characteristics of the limited proliferation of the MSCs of flatworms, set a fixed growth probability based on the number of disassembly tasks, and let the fixed number but randomly selected disassembly tasks replicate themselves to complete the growth process.

Figure 6 depicts the growth process of P16. Here, the growth probability is set to 0.1, then 2 of the 16 disassembly tasks will be randomly selected for growth. Assuming that tasks 9 and 15 are selected separately, then the available positions of task 9 can be inserted only to the immediate left of task 9 or 8 or 10, according to the disassembly precedence relationship of the tasks, assuming that the actual insertion position is to the immediate left of task 8. In the same way, the actual insertion position of task 15 is fixed to the immediate left of task 14. At this time, the disassembly task sequence is no longer feasible because of the existence of redundant tasks, but it is prepared for the next splitting process.

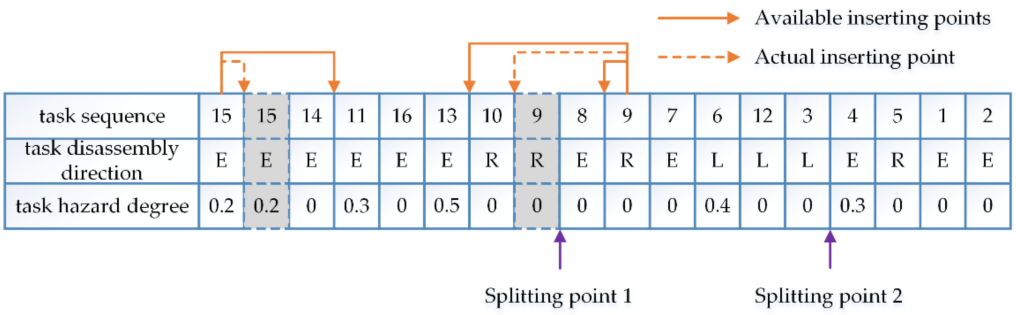

Figure 6. The growth process of 16 disassembly tasks.

3.3. The Splitting Process

Based on the fact that flatworms are accidentally divided into two or more segments when stimulated, the grown disassembly task sequence enters the splitting process. To increase the diversity of feasible solutions in the evolution process, the number of splitting points is the same as the number of growth tasks, but the splitting positions adopt a random mode.

Assume that the two splitting points in Figure 6 are immediate to the left of tasks 8 and 4, then the grown disassembly task sequence is split into three segments, as shown in Figure 7.

3.4. The Regeneration Process

The partial tissue of the split flatworm slowly regenerates the lost tissue in a suitable environment. Based on the same mechanism, the regeneration of the split disassembly task sequence begins to regenerate the lost disassembly tasks. The main steps are as follows: first, ensure that the disassembly task of each remaining split part is unique; then find all

the lost disassembly tasks; and finally, regenerate each lost disassembly task one by one according to the task disassembly precedence relationship.

task sequence part1	15	15	14	11	16	13	10	9									
task disassembly direction part1	E	E	E	E	E	E	R	R									
task hazard degree part1	0.2	0.2	0	0.3	0	0.5	0	0									
task sequence part2									8	9	7	6	12	3			
task disassembly direction part2									E	R	E	L	L	L			
task hazard degree part2									0	0	0	0.4	0	0			
task sequence part3															4	5	1	2
task disassembly direction part3															E	R	E	E
task hazard degree part3															0.3	0	0	0

Figure 7. The splitting process of 16 disassembly tasks.

Tasks with higher hazard degrees and later to be disassembled will consume more added energy; therefore, in the regeneration, priority should be given to removing high-hazard tasks to reduce the energy consumption, and meanwhile, the convergence can be accelerated to prevent the random search of solution space. Figure 8 depicts the full state of the three split disassembly task sequence segments after regeneration. Taking the regeneration process based on the intermediate segment as an example, first, it can be found that all tasks in the fragment are unique, and the lost task set is {1,2,4,5,8,10,11,13,14,15,16}; then, according to the disassembly precedence relationship, tasks 16, 15 and 14 can be the first candidate task in the new regenerated task sequence, and task 15 is selected since it has a hazard degree of 0.5 unit, then task 16 is randomly identified as the second, when tasks 14, 13, and 11 become available, task 13 is selected based on the same principle of hazard degree priority. Other lost disassembly tasks are selected into the new regenerated task sequence similarly.

task sequence 1	15	14	11	16	13	10	9	8	7	6	4	5	2	12	3	1
task disassembly direction 1	E	E	E	E	E	R	E	E	E	L	E	R	E	L	L	E
task hazard degree 1	0.2	0	0.3	0	0.5	0	0	0	0	0.4	0.3	0	0	0	0	0
task sequence 2	15	16	13	14	11	8	10	9	7	6	12	3	4	5	2	1
task disassembly direction 2	E	E	E	E	E	E	R	R	E	L	L	L	E	R	E	E
task hazard degree 2	0.2	0	0.5	0	0.3	0	0	0	0	0.4	0	0	0.3	0	0	0
task sequence 3	15	14	11	12	16	13	10	9	8	7	6	3	4	5	1	2
task disassembly direction 3	E	E	E	E	E	R	E	R	E	L	L	L	E	R	E	E
task hazard degree 3	0.2	0	0.3	0	0	0.5	0	0	0	0.4	0.4	0	0.3	0	0	0

Figure 8. The regeneration process of 16 disassembly tasks.

3.5. Solution Decoding

The process of assigning each task of disassembly task sequence to the workstation in precise order under the constraints of the disassembly direction, parallel operation and fuzzy cycle time is called decoding. A pair of tasks that need a parallel operation

should be disassembled at both left and right stations in the same mated-station and completed simultaneously.

The fuzzy cycle time is set to (15,18,21), two decoding schemes of P16 are depicted in Figures 9 and 10, where Figure 9 is a task assignment scheme of an initial encoding scheme in Figure 5, and Figure 10 is an assignment of the regeneration of the disassembly task sequence based on the intermediate segment Figure 8. Both 4 mated-stations are utilized, a total of 8 stations are utilized in Figure 9, while only 7 stations in 4 mated-stations are utilized in Figure 10, the parallel operation task pairs {11,12} and {5,6} are completed in mated-stations 1 and 3, respectively. Intuitively, compared with the first decoding scheme, the second decoding scheme not only uses fewer workstations to complete all disassembly tasks, but also causes less idle time on the disassembly line, which also leads to less energy consumption. This illustrates the effectiveness of the growth, splitting and regeneration process preliminarily, and detailed cases will be introduced to verify this next.

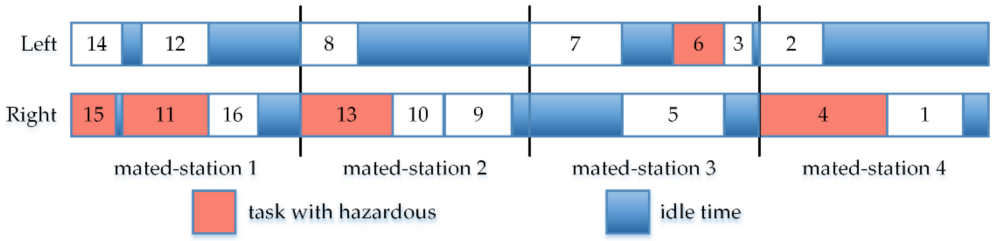

Figure 9. Decoding scheme of an initial encoding scheme.

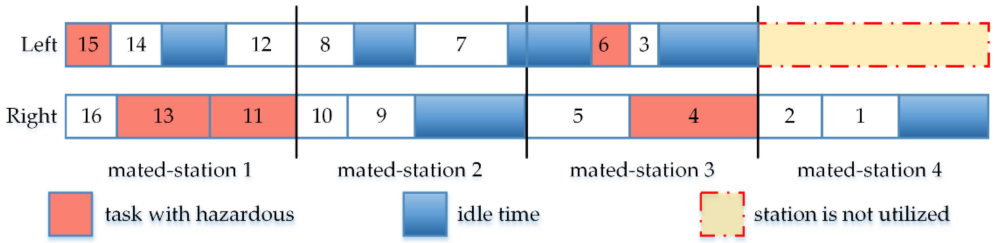

Figure 10. Decoding scheme of a regeneration individual.

3.6. The Pareto-Optimal Solutions

The Pareto-optimal solutions are introduced to compare the quality of the solutions of the multiobjective problem. Taking the multiobjective minimization problem as an example, there are three objectives in this paper. Assuming that there are two feasible solutions X1 and X2, if all the objectives in X1 are less than or equal to those in X2, and at least one objective in X1 is clearly less than the corresponding objective in X2, then we call X1 dominates X2. If the above conditions are not fully satisfied, then X1 and X2 are called nondominated solutions. Furthermore, if X1 and X2 are nondominated solutions, and no other feasible solution can dominate X1 and X2, then X1 and X2 are called Pareto-optimal solutions [8].

4. Case Verification and Discussion

The proposed MOFA is coded in the environment of Matlab R2019a on Windows 10 with a 3.2 GHz processor and 8 GB RAM. In the following section, three different types of cases show that the proposed model and MOFA can provide more exciting solutions for TDLBP-POF problems.

4.1. Fuzzy Straight One-Sided Disassembly Line Case with No Parallel Operation

Two scales of the straight disassembly line with fuzzy processing times are taken from Kalayci [30] to study the effectiveness of the proposed algorithm, one is the mobile phone case with 25 tasks and another is the laptop case with 47 tasks. Four main definite objectives

and three additional fuzzy objectives of the line balance efficiency ($\widetilde{BE}$), smoothness index ($\widetilde{SI}$) and the line balance delay time ($\widetilde{BD}$) are considered. Three Pareto-optimal solutions by the hybrid discrete artificial bee colony (HDABC) [30] and eight by the improved artificial fish swarm algorithm (IAFSA) [29] are found for the mobile phone case. The parameters of the proposed MOFA for the mobile phone case are set as follows: n =30, p = 0.05, $\widetilde{CT}$ = (12, 15, 18) and the total iteration is 50.

More than 10 Pareto-optimal solutions are obtained, of which 5 are listed in Table 1, and the task assignment scheme of each solution is shown in Table 2. The minimum values of four definite objectives both are (12, 14.048, 85, 919) by HDABC and IAFSA, while the minimums by MOFA are (11, 10.714, 77, 913). Partial obtained solutions by HDABC and IAFSA are dominated by the MOFA, which indicates that the MOFA is superior to the above two methods in obtaining more and better Pareto-optimal solutions for the mobile phone case.

Table 1. Computed results by the multiobjective flatworm algorithm (MOFA) for the mobile phone case.

Solution	f1	f2	f3	f4	$\widetilde{BE}$	$\widetilde{SI}$	$\widetilde{BD}$
1	11	10.714	79	908	(0.551, 0.891, 1)	(0, 7.616, 27.622)	(0, 18, 89)
2	11	10.960	75	908	(0.551, 0.891, 1)	(0, 8.000, 27. 839)	(0, 18, 89)
3	12	15.239	73	882	(0.505, 0.817, 1)	(3, 12.845, 32.265)	(0, 33, 107)
4	12	16.422	76	856	(0.505, 0.817, 1)	(3, 14.799, 33.091)	(3, 33, 107)
5	13	20.887	71	873	(0.466, 0.754, 1)	(9.055, 18.868, 36.756)	(14, 48, 125)

f1: number of workstations. f2: smoothness index. f3: disassembly priority of hazardous parts. f4: disassembly priority of high demand parts. $\widetilde{BE}$: fuzzy line balance efficiency. $\widetilde{SI}$: fuzzy smoothness index. $\widetilde{BD}$: fuzzy line balance delay time.

Table 2. Task assignment schemes by the multiobjective flatworm algorithm (MOFA) for the mobile phone case.

Solution	Task Assignment Schemes
1	{4}-{5,2}-{1,10,11,12,3}-{8}-{6}-{7}-{9}-{13,14,17,21,22}-{25,15,18,16,20}-{19}-{23,24}
2	{2,4}-{1,5}-{10,11,12,3}-{7}-{6}-{9}-{8}-{14,13,17,15,18,16}-{19,21}-{25,22,20}-{23,24}
3	{2,4}-{1,5}-{10,11,12,3}-{8}-{7}-{6}-{9}-{14,13,17,21,25}-{22,15,18,16}-{19}-{23}-{20,24}
4	{2,1}-{7}-{6}-{5,3}-{9}-{8}-{14,13,17,21,25}-{22,4}-{16,23}-{10,11,12,15,18}-{19}-{20,24}
5	{2,5}-{1,4}-{10,11,12,3}-{8}-{6}-{7}-{9}-{14,13,17,21,25}-{22,16}-{23,15}-{18}-{19}-{20,24}

The parameters for the laptop case are set as follows: n =50, P = 0.05, $\widetilde{CT}$ = (98, 104, 110) and the total iteration is set to 50. More than 20 Pareto-optimal solutions are obtained, of which 9 are listed in Table 3, while only 3 Pareto-optimal solutions by HDABC [30] and 10 by IAFSA [29] are found for the laptop case. The corresponding task assignment by the proposed MOFA is shown in Table 4. The minimum number of workstations and smoothness index by MOFA are 9 and 38.76, respectively, which are less than the computed results by HDABC and IAFSA, which also shows that the performance of the proposed MOFA can obtain more high-quality and diverse optimal solutions.

Table 3. Computed results by the multiobjective flatworm algorithm (MOFA) for the laptop case.

Solution	f_1	f_2	f_3	f_4	$\widetilde{BE}$	$\widetilde{SI}$	$\widetilde{BD}$
1	9	39.12	271	2372	(0.72, 0.91, 1)	(0, 30.56, 95.35)	(0, 80, 278)
2	9	39.96	263	2335	(0.72, 0.91, 1)	(0, 32.28, 95.27)	(0, 80, 278)
3	9	40.28	264	2325	(0.72, 0.91, 1)	(0, 32.77, 95.58)	(0, 80, 278)
4	9	45.43	261	2334	(0.72, 0.91, 1)	(9.00, 38.08, 96.58)	(9, 80, 278)
5	10	71.01	259	2330	(0.65, 0.82, 1)	(14.93, 69.70, 129.69)	(27, 184, 388)
6	10	74.00	261	2313	(0.65, 0.82, 1)	(21.42, 71.96, 130.66)	(33, 184, 388)
7	10	79.18	257	2322	(0.65, 0.82, 1)	(29.41, 77.06, 133.21)	(47, 184, 388)
8	11	129.08	260	2311	(0.59, 0.75, 0.97)	(93.77, 125.90, 170.76)	(134, 288, 498)
9	11	144.90	259	2303	(0.59, 0.75, 0.97)	(117.69, 141.63, 178.66)	(192, 288, 498)

f1: number of workstations. f2: smoothness index. f3: disassembly priority of hazardous parts. f4: disassembly priority of high demand parts. $\widetilde{BE}$: fuzzy line balance efficiency. $\widetilde{SI}$: fuzzy smoothness index. $\widetilde{BD}$: fuzzy line balance delay time.

Table 4. Task assignment schemes by the multiobjective flatworm algorithm (MOFA) for the laptop case.

Solution	Task Assignment Schemes
1	{5,13,14,19,15,16,17,18,47}-{20,21,22,23,24}-{25,8,9,10,11,1,6}-{26,27}-{2,3,28,29,30,31,4}-{32}-{33,34,35,36,43,44,12,45}-{37,46,7,38}-{39,40,41,42}
2	{5,13,14,19,15,16,17,18,47}-{20,22,21,24,23}-{25,8,9,10,11,1}-{26}-{2,3,27,28,29,30,31}-{32}-{33,34,35,36,43,44,45,46,6}-{12,7,37,38,4}-{39,40,41,42}
3	{5,13,14,19,15,16,17,18,47}-{20,22,21,24,23}-{25,8,9,10,11,1}-{26,27}-{28,29,30,31,2,3}-{32}-{33,34,35,36,43,44,45,46,12}-{6,7,37,38,4}-{39,40,41,42}
4	{5,13,14,19,15,16,17,18,47,22}-{20,21,23,24,25}-{1,2,3,8,9}-{26}-{10,11,27,28,29,30,31}-{32}-{33,34,35,36,43,44,45,46,12}-{6,7,37,38,4}-{39,40,41,42}
5	{5,13,14,19,15,16,17,18,47,22}-{20,21,23,24,25}-{1,2,3,8,9}-{26}-{10,11,27,28,29,30,31}-{32}-{33,34,35,36,43,44,6}-{37,38}-{39,40}-{45,46,12,41,42,4,7}
6	{5,13,14,19,15,16,17,18,47,22}-{20,21,23,24,25}-{8,9,10,11,1}-{26,27}-{28,29,30,31,2,3}-{32}-{33,34,35,36,43,44}-{37,38}-{39,40,6}-{45,46,12,41,42,4,7}
7	{5,13,14,19,15,16,17,18,47,22}-{20,21,23,24,25}-{1,2,3,8,10}-{9,11,26}-{27,28,29,30,31}-{32}-{33,34,35,36,43,44}-{37,38}-{39,40}-{45,46,12,41,42,6,4,7}
8	{5,13,14,19,15,16,17,18,47,22}-{20,21,23,24,25}-{8,9,10,11}-{26,27}-{28,29,1,30,31,2}-{3}-{32}-{33,34,35,36,43,44,6}-{37,38}-{39,40}-{45,46,12,41,42,4,7}
9	{5,13,14,19,15,16,17,18,47,22}-{20,21,23,24,25}-{8,9,10,11}-{26,27}-{28,29,1,30,31,2}-{3}-{32}-{33,34,35,36,43,44,45,46,12}-{37,38}-{39,40,41,42}-{6,4,7}

4.2. Fuzzy Two-Sided Disassembly Line with Parallel Operation Tasks

To illustrate the performance of the proposed MOFA in the two-sided disassembly line with parallel operation tasks, the disassembly model and data in this section are modified from Figure 2. Meanwhile, to enrich the diversity of validation model data, the α-cut sets are introduced to produce more triangular fuzzy processing times; α is computed by Equation (23).

$$\alpha = \mu_{\widetilde{A}}(x) = \begin{cases} x - A^l / A^m - A^l, \ A^l \le x < A^m \\ A^u - x / A^u - A^m, \ A^m \le x < A^u \\ 0, \ otherwise \end{cases} \quad (23)$$

Let the fuzzy processing times in Figure 2 be the initial times when $\alpha = 0$, and the fuzzy task times of different α-cut sets can be obtained by Equation (23) in Table 5. Parameter α is set to 0, 0.3, 0.5, 0.7 and 1, respectively. The fuzzy degree of the task times gradually decreases with the increase of α; in particular, the fuzzy task times become a deterministic value when $\alpha = 1$, which indicates $A^l = A^m = A^u$ and is abbreviated as a single number.

Two other multiobjective algorithms PSO and the improved nondominated sorting genetic algorithm (NSGA-II) based on the population optimization are designed to test the performance of the proposed MOFA. The parameters of the three algorithms are set as n =16, $\widetilde{CT} = (15, 18, 21)$ and the maximum iteration is set to 30. The inertia weight, personal and global learning coefficients of PSO are set to 0.6, 1, and 2. The crossover and mutation probabilities of NSGA-II are set to 0.8 and 0.2. Other parameters are set as $\eta = 0.6$, $e_{ft} = 1$, $e_{eq} = 1$, $e_h = 0.2$. Each algorithm with different levels of α is executed 10 times; the computed results are shown in Tables 6–8.

To facilitate analysis, the results of different α are converted to real numbers for comparison, and the scatter point distribution in the three-dimensional space is shown in Figure 11. On the whole, three objectives showed a downward trend. The reason is that as α increases, the uncertainty in the disassembly process decreases, which is in accord with the actual situation that the fuzzier the task processing time, the worse the overall balance effect. Meanwhile, as shown in Figure 11, only one solution obtained by MOFA has higher energy consumption; the others are significantly lower than the other two algorithms. When α is equal to 1, the minimum energy consumption of MOFA, NSGA-II and PSO is

149.26, 149.35 and 149.28, respectively, considering the balance loss rate and smoothness index; obviously, MOFA is superior to the other two algorithms, and the Pareto-optimal solutions (8.89, 16.00,149.37) and (8.89, 20.00,149.31) obtained by MOFA even dominate the other two algorithms. The superiority of the proposed MOFA to the comprehensive optimization of the disassembly line is proved.

Table 5. Fuzzy task processing time with different fuzzy degrees of P16.

Task	$\alpha = 0$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 1$
1	(4, 6, 8)	(4.6, 6, 7.4)	(5, 6, 7)	(5.4, 6, 6.6)	6
2	(4, 5, 6)	(4.3, 5, 5.7)	(4.5, 5, 5.5)	(4.7, 5, 5.3)	5
3	(1, 2, 4)	(1.3, 2, 3.4)	(1.5, 2, 3)	(1.7, 2, 2.6)	2
4	(8, 9, 14)	(8.3, 9, 12.5)	(8.5, 9, 11.5)	(8.7, 9, 10.5)	9
5	(6, 8, 10)	(6.6, 8, 9.4)	(7, 8, 9)	(7.4, 8, 8.6)	8
6	(2, 4, 6)	(2.6, 4, 5.4)	(3, 4, 5)	(3.4, 4, 4.6)	4
7	(5, 7, 10)	(5.6, 7, 9.1)	(6, 7, 8.5)	(6.4, 7, 7.9)	7
8	(2, 4, 8)	(2.6, 4, 6.8)	(3, 4, 6)	(3.4, 4, 5.2)	4
9	(3, 5, 8)	(3.6, 5, 7.1)	(4, 5, 6.5)	(4.4, 5, 5.9)	5
10	(2, 4, 6)	(2.6, 4, 5.4)	(3, 4, 5)	(3.4, 4, 4.6)	4
11	(5, 6, 10)	(5.3, 6, 8.8)	(5.5, 6, 8)	(5.7, 6, 7.2)	6
12	(4, 5, 8)	(4.3, 5, 7.1)	(4.5, 5, 6.5)	(4.7, 5, 5.9)	5
13	(5, 6, 12)	(5.3, 6, 10.2)	(5.5, 6, 9)	(5.7, 6, 7.8)	6
14	(2, 4, 6)	(2.6, 4, 5.4)	(3, 4, 5)	(3.4, 4, 4.6)	4
15	(2, 3, 6)	(2.3, 3, 5.1)	(2.5, 3, 4.5)	(2.7, 3, 3.9)	3
16	(2, 4, 6)	(2.6, 4, 5.4)	(3, 4, 5)	(3.4, 4, 4.6)	4

α: Different fuzzy degrees.

Table 6. Results of different fuzzy degrees by multiobjective particle swarm optimization (PSO).

α	$\widetilde{BLR}$	$\widetilde{SI}$	$\widetilde{TEC}$
0	(0, 8.89, 63.46)	(0, 16.00, 1643.00)	(113.01, 149.37, 207.97)
	(0, 16.33, 59.29)	(0, 74.00, 1023.00)	(127.99, 167.33, 228.91)
	(0, 21.90, 59.29)	(0, 109.00, 997.00)	(127.96, 167.29, 228.86)
0.3	(0, 8.89, 52.64)	(0, 16.00, 863.21)	(123.92, 149.37, 190.39)
	(0, 21.15, 46.61)	(12.25, 96.00, 422.41)	(155.82, 185.48, 230.77)
	(0, 19.61, 53.86)	(0, 106.00, 969.59)	(123.84, 149.28, 190.27)
0.5	(0, 8.89, 43.50)	(0, 16.00, 482.25)	(131.19, 149.37, 178.67)
	(0, 14.58, 42.08)	(1, 66.00, 449.75)	(131.11, 149.28, 178.56)
0.7	(0, 8.89, 34.30)	(0.18, 20, 264.29)	(138.41, 149.31, 166.89)
	(0, 14.58, 32.52)	(3.38, 46.00, 233.35)	(138.38, 149.28, 166.85)
1	14.58	42.00	149.33
	14.58	46.00	149.31
	14.58	66.00	149.28

α: different fuzzy degrees. $\widetilde{BLR}$: fuzzy line balance loss rate. $\widetilde{SI}$: fuzzy smoothness index. $\widetilde{TEC}$: fuzzy energy consumption.

4.3. Application Case Verification and Analysis

This section will take a large-capacity automobile disassembly line in Chongqing, China, as the research application; the layout of the automobile disassembly line is shown in Figure 12, involving 62 disassembly tasks. Due to the symmetry of the assembly position of automobile parts, 19 tasks are disassembled on the left, 15 tasks are on the right, and the rest can be performed on the left or right side of the line. During the disassembly process, 7 pairs of 14 disassembly tasks need to be disassembled in parallel, marked as TPO, such as tasks 1 and 2. Only when the two disassembly tasks are completed on both sides of the same mated-station on the line at the same time, task 3 can be performed. The fuzzy disassembly time is obtained by the continuous measurement method of the stopwatch, and the degree of disassembly hazard is obtained from historical disassembly statistics. The fuzzy disassembly cycle time is set to (480,510,550). Details are shown in Table 9.

Table 7. Results of different fuzzy degrees by the improved nondominated sorting genetic algorithm (NSGA-II).

α	$\widetilde{BLR}$	$\widetilde{SI}$	$\widetilde{TEC}$
0	(0, 14.58, 60.42)	(0, 56.00, 1275.00)	(113.06, 149.43, 208.09)
	(0, 14.58, 63.46)	(0, 48.00, 1647.00)	(112.97, 149.31, 207.89)
	(0, 21.90, 59.29)	(0, 99.00, 995.00)	(128.02, 167.37, 229.00)
0.3	(0, 14.58, 50.23)	(0, 56.00, 720.51)	(123.91, 149.35, 190.38)
	(0, 14.58, 53.26)	(0, 48.00, 912.99)	(123.88, 149.31, 190.32)
	(0, 21.90, 50.19)	(0, 101.00, 617.32)	(139.92, 167.48, 210.67)
	(0, 21.90, 50.19)	(0, 101.00, 617.32)	(139.92, 167.48, 210.67)
0.5	(0, 8.89, 43.50)	(0, 18.00, 483.75)	(131.29, 149.48, 178.82)
	(0, 14.58, 42.08)	(4.00, 66.00, 458.75)	(131.21, 149.39, 178.71)
	(0, 14.58, 47.35)	(0, 52.00, 666.75)	(131.21, 149.39, 178.70)
	(0, 19.61, 46.12)	(0.25, 84.00, 606.75)	(131.17, 149.35, 178.66)
0.7	(0, 8.89, 32.15)	(0, 18.00, 214.23)	(152.16, 163.78, 182.09)
	(0, 14.58, 32.52)	(2.18, 44.00, 227.39)	(138.45, 149.35, 166.93)
	(0, 14.58, 32.52)	(14.89, 66.00, 246.75)	(138.37, 149.28, 166.84)
1	8.89	24.00	163.89
	14.58	44.00	149.38
	19.61	84.00	149.35

α: different fuzzy degrees. $\widetilde{BLR}$: fuzzy line balance loss rate. $\widetilde{SI}$: fuzzy smoothness index. $\widetilde{TEC}$: fuzzy energy consumption.

Table 8. Computed results of different fuzzy degrees by the multiobjective flatworm algorithm (MOFA).

α	$\widetilde{BLR}$	$\widetilde{SI}$	$\widetilde{TEC}$
0	(0, 8.89, 60.42)	(0, 22.00, 1277.00)	(125.04, 163.80, 224.84)
	(0, 8.89, 63.46)	(0, 16.00, 1643.00)	(113.01, 149.37, 207.97)
	(0, 8.89, 66.07)	(0, 27.00, 2085.00)	(112.93, 149.26, 207.81)
	(0, 14.58, 60.42)	(0, 46.00, 1275.00)	(112.99, 149.33, 207.93)
	(0, 16.33, 59.29)	(0, 74.00, 1023.00)	(127.99, 167.33, 228.91)
0.3	(0, 8.89, 52.64)	(0, 16.00, 863.21)	(123.92, 149.37, 190.39)
	(0, 8.89, 55.39)	(0, 20, 1088.69)	(123.88, 149.31, 190.32)
	(0, 14.58, 50.23)	(0, 42.00, 720.79)	(123.89, 149.33, 190.34)
0.5	(0, 8.89, 37.39)	(0, 18.00, 294.75)	(131.20, 149.38, 178.69)
	(0, 8.89, 43.50)	(0, 16.00, 482.25)	(131.19, 149.37, 178.67)
	(0, 14.58, 42.08)	(0, 42.00, 437.75)	(131.16, 149.33, 178.63)
	(0, 14.58, 42.08)	(0, 46.00, 447.75)	(131.11, 149.28, 178.56)
0.7	(0, 8.89, 27.39)	(0.04, 18.00, 138.75)	(138.47, 149.38, 166.96)
	(0, 8.89, 32.15)	(0, 16.00, 212.41)	(138.46, 149.37, 166.95)
	(0, 8.89, 34.30)	(0.18, 20, 264.29)	(138.41, 149.31, 166.89)
	(0, 14.58, 32.52)	(3.38, 46.00, 233.35)	(138.38, 149.28, 166.85)
	(0, 14.58, 34.65)	(1.37, 50, 278.93)	(138.36, 149.26, 166.82)
1	8.89	16.00	149.37
	8.89	20.00	149.31
	14.58	44.00	149.27
	14.58	50.00	149.26

α: different fuzzy degrees. $\widetilde{BLR}$: fuzzy line balance loss rate. $\widetilde{SI}$: fuzzy smoothness index. $\widetilde{TEC}$: fuzzy energy consumption.

Before the algorithm comparison test, the parameters of each algorithm need to be calibrated. The orthogonal test design and the analysis of variance (ANOVA) techniques are utilized to determine the value of each parameter in this paper. Taking the proposed MOFA as an example, all core parameters, namely the population number (n: 60, 80 and 100), the growth probability (p: 0.05, 0.07 and 0.1), and the total number of evolutionary iterations (s: 120, 160 and 200) are tested at three levels. The three main parameters and the interaction between each two parameters are considered, and the simultaneous interaction between the three is ignored. The energy consumption is selected as the dependent variable. Repeated

experiments were performed to improve the reliability of the statistical analysis, so the energy consumption was measured three times under each same factor level configuration, and the significance level was set at 0.05. The result of the analysis of variance for MOFA is shown in Table 10. Obviously, except for factors n and s and their interactions that are considered to be highly statistically significant, other factors are not highly statistically significant. Therefore, the preferred parameters of n and s are determined as 80 and 160, after the estimated marginal mean values of the energy consumption with all available configurations, and parameter p can be set to any value in all available levels; here it is set to 0.05.

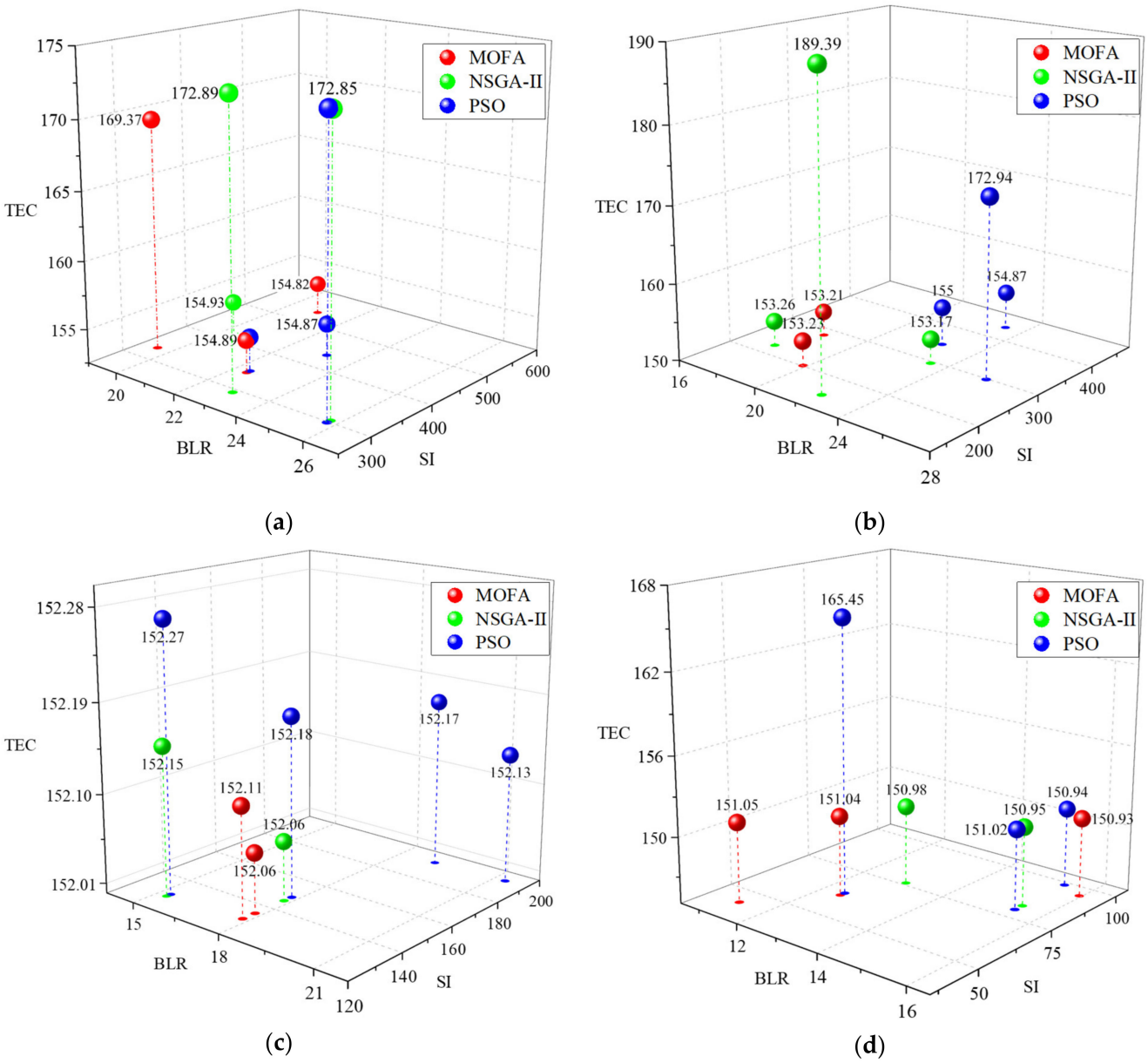

Figure 11. Distributions of crisp real number Pareto-optimal solutions of three algorithms: (**a**) fuzzy degree $\alpha = 0$; (**b**) fuzzy degree $\alpha = 0.3$; (**c**) fuzzy degree $\alpha = 0.5$; (**d**) fuzzy degree $\alpha = 0.7$.

Figure 12. Site of the automobile two-sided disassembly line.

Table 9. The automobile disassembly data information with 62 tasks.

Task	Description	$\widetilde{t}$	h	K	TPO	S
1	Engine cover left bolt	(30,36,41)	0	L	2	3
2	Engine cover right bolt	(30,36,41)	0	R	1	3
3	Engine cover	(22,30,36)	0	E		5,7
4	Air bag	(160,185,200)	1	L		
5	Battery	(66,73,85)	0.8	R		6
6	Waste fluid	(151,162,190)	0.5	E		
7	Waste oil	(200,225,252)	0.7	E		
8	Left wheels	(112,128,136)	0	L		10
9	Right wheels	(112,128,136)	0	R		11
10	Left fender	(50,56,65)	0	L		12
11	Right fender	(50,56,65)	0	R		13
12	Left front bumper	(42,49,57)	0	L	13	14
13	Right front bumper	(42,49,57)	0	R	12	14
14	Front bumper	(30,34,41)	0	E		15,16,17
15	Air intake grille	(41,46,53)	0	E		26,27,28,29
16	Left mirrors and lamps	(55,70,79)	0	L		
17	Right mirrors and lamps	(55,70,79)	0	R		
18	Left door	(107,127,139)	0	L		37
19	Right door	(87,97,109)	0	R		38
20	Left trunk cover hinge	(22,25,30)	0	L	21	22
21	Right trunk cover hinge	(22,25,30)	0	R	20	22
22	Trunk cover	(31,36,42)	0	E		
23	Left behind bumper	(55,61,72)	0	L	24	25
24	Right behind bumper	(55,61,72)	0	R	23	25
25	Behind bumper	(23,26,35)	0	E		16,17
26	Radiator	(90,97,110)	0	E		56
27	Condenser	(101,106,121)	0	E		56
28	Coolant tank	(103,112,127)	0.3	E		56

Table 9. *Cont.*

Task	Description	$\widetilde{t}$	*h*	*K*	*TPO*	*S*
29	Air cleaner	(57,63,70)	0	E		56
30	Wiper	(47,51,59)	0	E		33,36
31	Front glass left sealant	(60,67,75)	0	L	32	33
32	Front glass right sealant	(60,67,75)	0	R	31	33
33	Front glass	(32,39,46)	0.3	E		41,44
34	Behind glass left sealant	(40,47,57)	0	L	35	36
35	Behind glass right sealant	(40,47,57)	0	R	34	36
36	Behind glass	(25,28,35)	0.2	E		
37	Left seat	(201,225,246)	0	L		39
38	Right seat	(200,216,235)	0	R		39
39	Armrest box	(90,98,117)	0	E		41
40	Fuel tank	(95,106,121)	0.2	R		
41	Steering wheel	(120,132,150)	0	L		45
42	Dashboard left bolt	(94,102,110)	0	L	43	44
43	Dashboard right bolt	(69,75,86)	0	R	42	44
44	Dashboard	(70,75,85)	0	E		45
45	Instrument	(87,98,110)	0	L		48
46	Shift handle	(105,121,139)	0	E		47
47	Brake rigging	(70,78,88)	0	L		48
48	Clutch pedal	(59,65,78)	0	L		49
49	Accelerator pedal	(55,60,72)	0	L		61
50	Air conditioner	(125,138,153)	0	E		51
51	Carbon canister	(35,41,50)	0.4	E		61,62
52	Bottom guard board	(56,61,77)	0	E		53
53	Ternary catalysis	(101,120,136)	0.7	E		54
54	Exhaust pipe	(104,118,127)	0.5	E		55
55	Transmission shaft	(230,258,280)	0	E		56
56	Electric generator	(124,135,154)	0	E		57
57	Engine	(450,475,499)	0	E		58
58	Transmission	(207,228,244)	0	E		59
59	Front suspension	(221,239,256)	0	E		60
60	Behind suspension	(229,248,268)	0	E		61,62
61	Left body accessories	(129,138,152)	0.1	L		
62	Right body accessories	(100,117,131)	0.1	R		

$\widetilde{t}$: task fuzzy disassembly time. *h*: task hazard degree. *K*: task disassembly direction. *TPO*: task with a parallel operation. *S*: immediate disassembly successors.

NSGA-II and the multiobjective PSO (MOPSO), differential evolution (MODE), and simulated annealing (MOSA) are selected to be compared against the MOFA. Based on the same orthogonal test design and ANOVA, the crossover and mutation rates of NSGAII and MODE are both set to 0.75 and 0.25. The inertia weight, personal and global learning coefficients of MOPSO are set to 0.6, 1 and 2. The end time of the total number of iterations of the algorithms is regarded as the end time of MOSA, and the initial temperature and cooling rate are set to 1000 and 0.98, respectively. Each algorithm is executed 10 times; a total of 47 crisp real number of Pareto-optimal solutions are obtained by MOFA, 17, 8, 15 and 11 Pareto-optimal solutions by NSGA-II, MOPSO, MODE and MOSA, respectively, by combining the 98 solutions together and recalculating the dominating relationship between them, 47 approximate true Pareto-optimal solutions (ATP) can be obtained. The distribution of Pareto-optimal solutions computed by each algorithm on each objective is shown in Figure 13.

It is clear from Figure 13 that the Pareto-optimal solutions obtained by MOFA can obtain a better distribution on any objective than other algorithms. Although the line balance loss rate (BLR) and smoothness index (SI) in Figure 13a,b are relatively higher, referring to Figure 13d, it can be seen that the corresponding energy consumption is relatively lower.

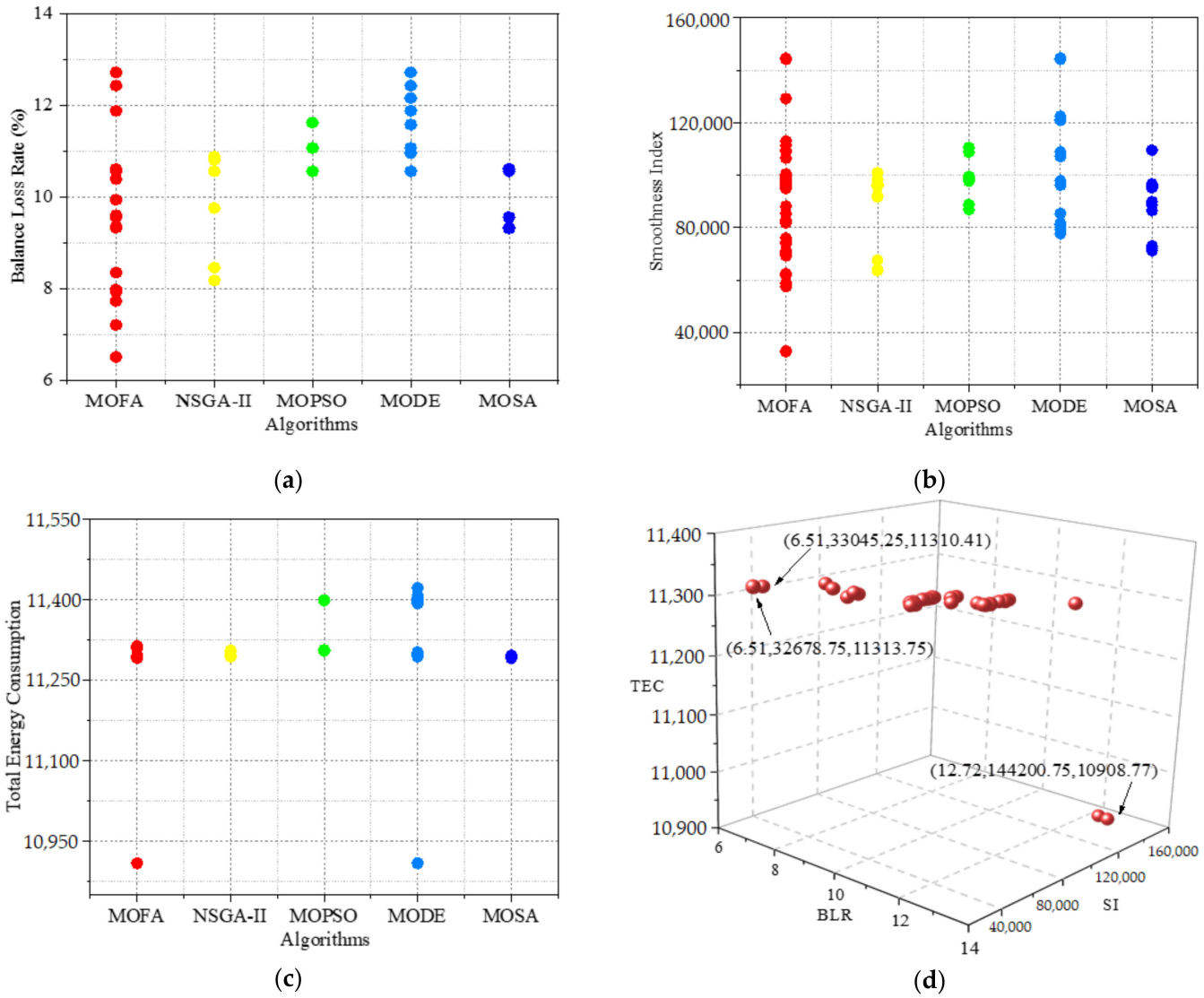

Figure 13. Distributions of the crisp real number of Pareto-optimal solutions of five algorithms: (**a**) balance loss rate; (**b**) smoothness index; (**c**) total energy consumption; (**d**) approximate true Pareto-optimal solutions (ATP) of five algorithms.

Three Pareto-optimal disassembly schemes of the proposed MOFA are depicted in Figures 14–16. First of all, both the minimum BLR and minimum SI schemes need to open 8 mated-stations, and in which 2 stations are not opened, while in the minimum TEC scheme only needs to open 7 mated-stations and each workstation is fully utilized. Secondly, the task pairs requiring parallel disassembly on both sides are satisfied, in the minimum BLR and minimum SI schemes; those operations are mainly completed on mated-station 2, while in the minimum TEC scheme are mainly operated on mated-station 1. Finally, the BLR of the minimum BLR and minimum SI schemes is 6.51%, and the SI and TEC are 32,678.75; 11,313.75 and 33,045.25; 11,310.41, respectively. While the minimum TEC scheme only needs 10,908.77 owning to the less energy consumption of lighting and ventilation by opening the fewer mated-stations. More importantly, the minimum TEC scheme can complete all hazardous tasks except task 7 before mated-station 4, which reduces the additional energy

consumption needed to deal with hazardous tasks, while the remaining two schemes can only complete most of the hazardous tasks before mated-station 5. However, it should be noted that the minimum TEC scheme leads to higher BLR and SI, which are 12.72% and 144,200.75, respectively.

Table 10. The ANOVA result for the parameters of MOFA.

Source	Type III Sum of Squares	df	Mean Square	F-Ratio	*p*-Value
Corrected Model	337312.773 [a]	18	18739.598	4.341	7.45×10^{-6}
Intercept	10307942906.242	1	10307942906	2388057.884	7.05×10^{-145}
n	78530.084	2	39265.042	9.097	3.43×10^{-4}
s	48637.008	2	24318.504	5.634	0.006
p	17303.642	2	8651.821	2.004	0.143
n ·s	130419.706	4	65209.854	15.107	0.002
n ·p	28153.932	4	14076.966	3.261	0.435
s ·p	34268.4	4	17134.199	3.969	0.292
Error	267620.171	62	4316.454		
Total	10308547839	81			
Corrected Total	604932.944	80			

[a] R squared = 0.558 (adjusted R squared = 0.429). n: population number. s: total evolutionary iterations. p: growth probability. n·s: interaction between n and s. n·p: interaction between n and p. s·p: interaction between s and p.

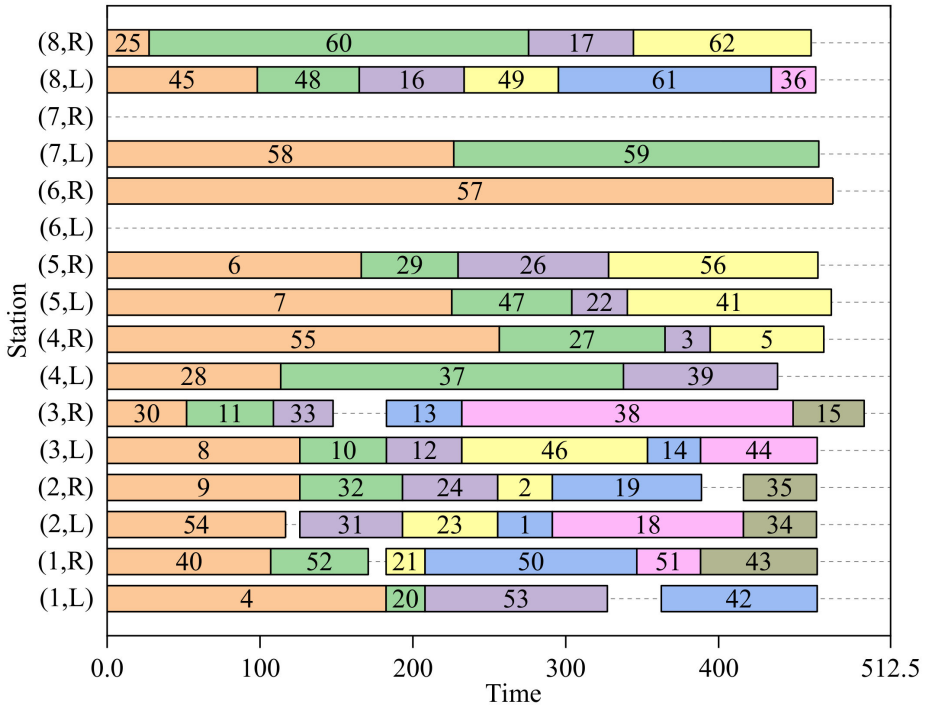

Figure 14. Minimum balance loss rate (*BLR*) scheme of the automobile disassembly.

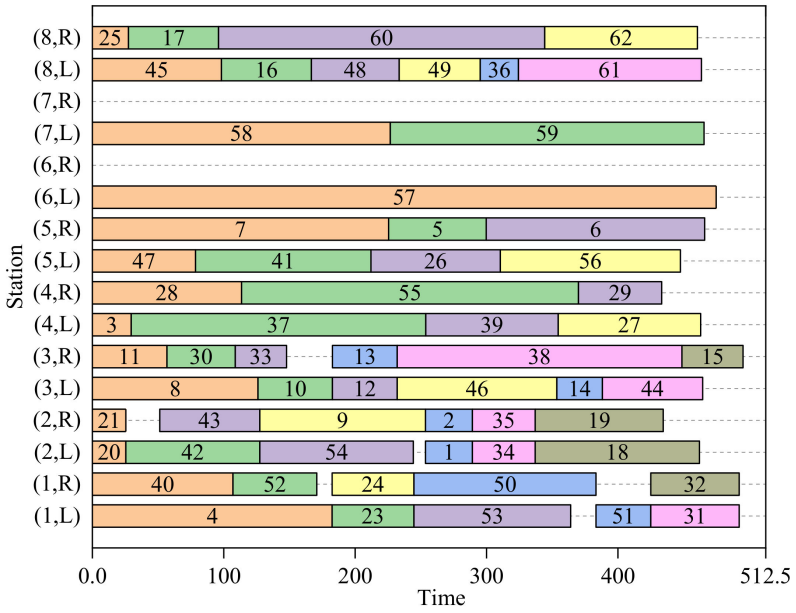

Figure 15. Minimum smoothness index (*SI*) scheme of the automobile disassembly.

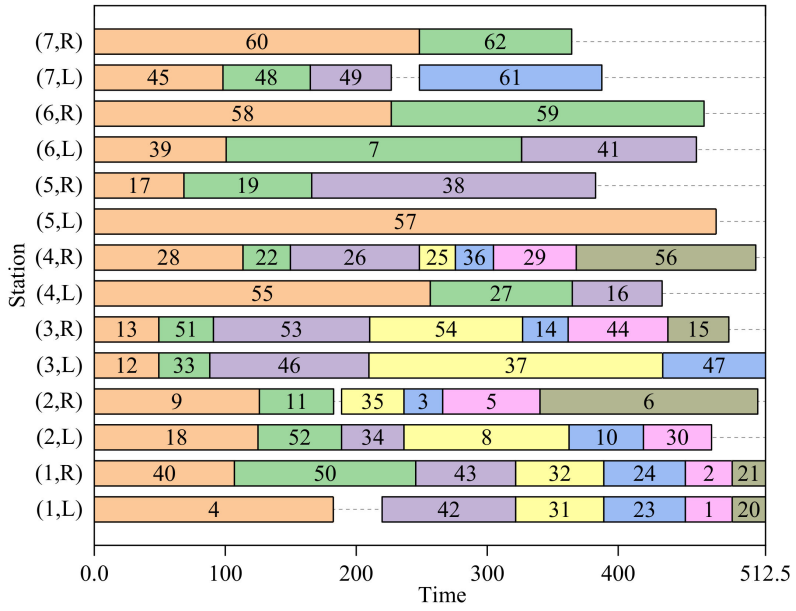

Figure 16. Minimum total energy consumption (*TEC*) scheme of the automobile disassembly.

5. Conclusions

In this paper, the authors propose an energy-efficient optimization model for the two-sided disassembly line. The model mainly considers the common but neglected operational constraints that some large-volume components or parts lay across the disassembly line should be dismantled parallel in the same mated-station under uncertain conditions. Three objectives of the disassembly line balance loss rate, smoothness index and energy consumption during the disassembly activities are designed. Then a multiobjective flatworm algorithm is proposed, and two existing literature cases and a self-designed case are designed to verify the effectiveness of the model and algorithm. Finally, the model and algorithm are applied to an automobile two-sided disassembly line. Results of the experiments show that, compared with NSGA-II, MOPSO, MODE and MOSA, the model and algorithm can provide a set of the higher quality Pareto-optimal solutions for disassembly scheme planners, enable disassembly workers to complete tasks in a more energy-effective manner

under the parallel operation constraint. Meanwhile, the proposed model and algorithm can reduce the line balance loss rate and workload smoothness during the disassembly process, and the decision-maker can choose the preferred solution from the multiobjective optimization solution according to the disassembly needs.

In future research, researchers can further improve the model to make it more consistent with the actual situation. At the same time, researchers can also study how to reduce the uncertainty in the demolition process and liberate the workers from the heavy disassembly operation, and use more efficient ways such as robots or human–robot collaborative disassembly, which are very valuable, and ultimately promote the green and sustainable development of EOL products disassembly industry.

Author Contributions: Conceptualization, J.L. and S.Z.; methodology, J.L. and S.G.; software, J.L.; validation, Y.Z., W.L. and J.L.; formal analysis, J.L.; investigation, S.G.; resources, S.Z.; data curation, Y.Z.; writing—original draft preparation, J.L.; writing—review and editing, S.G.; visualization, Y.Z.; supervision, S.G.; project administration, W.L.; funding acquisition, S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant number 51705386) and the Hubei Province Natural Science Foundation of China (grant number 2017CFB297). The APC was funded by Key Laboratory of Colleges and Universities in Sichuan Province for Enterprise Informationization and Internet of Things Measurement and Control Technology (grant number 2016WYY02) and Key Laboratory of Colleges and Universities in Sichuan Province for Non-destructive Testing of Bridges and Engineering Calculations (grant number 2016QYY05), Safety Production Technology Project of Sichuan Safety Supervision Bureau and Sichuan Coal Mine Safety Supervision Bureau (grant number aj20170601105926).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Acknowledgments: Thanks to all the reference authors who gave us inspiration and help. The authors thank the editors and anonymous commentators for their valuable comments, which have improved the quality of this paper, and the experimental institutions that provide financial support for the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xia, X.; Liu, W.; Zhang, Z.; Wang, L.; Cao, J.; Liu, X. A balancing method of mixed-model disassembly line in random working environment. *Sustainability* **2019**, *11*, 2304. [CrossRef]
2. Meng, W.; Zhang, X. Optimization of remanufacturing disassembly line balance considering multiple failures and material hazards. *Sustainability* **2020**, *12*, 7318. [CrossRef]
3. Kazancoglu, Y.; Ozturkoglu, Y. Integrated framework of disassembly line balancing with Green and business objectives using a mixed MCDM. *J. Clean. Prod.* **2018**, *191*, 179–191. [CrossRef]
4. Baykasoglu, A.; Tasan, S.O.; Tasan, A.S.; Akyol, S.D. Modeling and solving assembly line design problems by considering human factors with a real-life application. *Hum. Factors Ergon. Manuf.* **2017**, *27*, 96–115. [CrossRef]
5. Wang, K.; Li, X.; Gao, L.; Garg, A. Partial disassembly line balancing for energy consumption and profit under uncertainty. *Robot. Comput. Integr. Manuf.* **2019**, *59*, 235–251. [CrossRef]
6. Kalaycilar, E.G.; Azizoglu, M.; Yeralan, S. A disassembly line balancing problem with fixed number of workstations. *Eur. J. Oper. Res.* **2016**, *249*, 592–604. [CrossRef]
7. Liang, J.; Guo, S.; Du, B.; Li, Y.; Guo, J.; Yang, Z.; Pang, S. Minimizing energy consumption in multi-objective two-sided disassembly line balancing problem with complex execution constraints using dual-individual simulated annealing algorithm. *J. Clean. Prod.* **2020**, *284*, 125418. [CrossRef]
8. Zhu, L.; Zhang, Z.; Wang, Y. A Pareto firefly algorithm for multi-objective disassembly line balancing problems with hazard evaluation. *Int. J. Prod. Res.* **2018**, *56*, 7354–7374. [CrossRef]
9. Wang, K.; Li, X.; Gao, L. A multi-objective discrete flower pollination algorithm for stochastic two-sided partial disassembly line balancing problem. *Comput. Ind. Eng.* **2019**, *130*, 634–649. [CrossRef]
10. Gupta, S.M.; Gungor, A.; Pochampally, K.; Kamarthi, S.V. Complications in disassembly line balancing. *Proc. SPIE Int. Conf. Environ. Concious Manuf.* **2000**, *4193*, 289–298.

11. McGovern, S.M.; Gupta, S.M. Combinatorial optimization analysis of the unary NP-complete disassembly line balancing problem. *Int. J. Prod. Res.* **2007**, *45*, 4485–4511. [CrossRef]
12. Edis, E.B.; Ilgin, M.A.; Edis, R.S. Disassembly line balancing with sequencing decisions: A mixed integer linear programming model and extensions. *J. Clean. Prod.* **2019**, *238*, 117826. [CrossRef]
13. Altekin, F.T. A comparison of piecewise linear programming formulations for stochastic disassembly line balancing. *Int. J. Prod. Res.* **2017**, *55*, 7412–7434. [CrossRef]
14. Özceylan, E.; Paksoy, T. Reverse supply chain optimisation with disassembly line balancing. *Int. J. Prod. Res.* **2013**, *51*, 5985–6001. [CrossRef]
15. Kannan, D.; Garg, K.; Jha, P.C.; Diabat, A. Integrating disassembly line balancing in the planning of a reverse logistics network from the perspective of a third party provider. *Ann. Oper. Res.* **2017**, *253*, 353–376. [CrossRef]
16. Cao, J.; Xia, X.; Wang, L.; Zhang, Z.; Liu, X. A Novel Multi-Efficiency Optimization Method for Disassembly Line Balancing Problem. *Sustainability* **2019**, *11*, 6969. [CrossRef]
17. Li, J.; Chen, X.; Zhu, Z.; Yang, C.; Chu, C. A branch, bound, and remember algorithm for the simple disassembly line balancing problem. *Comput. Oper. Res.* **2019**, *105*, 47–57. [CrossRef]
18. Li, Z.; Çil, Z.A.; Mete, S.; Kucukkoc, I. A fast branch, bound and remember algorithm for disassembly line balancing problem. *Int. J. Prod. Res.* **2020**, *58*, 3220–3234. [CrossRef]
19. Seidi, M.; Saghari, S. The balancing of disassembly line of automobile engine using Genetic Algorithm (GA) in fuzzy environment. *Ind. Eng. Manag. Syst.* **2016**, *15*, 364–373. [CrossRef]
20. Ding, L.P.; Feng, Y.X.; Tan, J.R.; Gao, Y.C. A new multi-objective ant colony algorithm for solving the disassembly line balancing problem. *Int. J. Adv. Manuf. Technol.* **2010**, *48*, 761–771. [CrossRef]
21. Kalayci, C.B.; Gupta, S.M. A particle swarm optimization algorithm with neighborhood-based mutation for sequence-dependent disassembly line balancing problem. *Int. J. Adv. Manuf. Technol.* **2013**, *69*, 197–209. [CrossRef]
22. Fang, Y.; Ming, H.; Li, M.; Liu, Q.; Pham, D.T. Multi-objective evolutionary simulated annealing optimisation for mixed-model multi-robotic disassembly line balancing with interval processing time. *Int. J. Prod. Res.* **2019**, *58*, 1–17. [CrossRef]
23. Liu, J.; Wang, S. Balancing disassembly line in product recovery to promote the coordinated development of economy and environment. *Sustainability* **2017**, *9*, 309. [CrossRef]
24. Ren, Y.; Yu, D.; Zhang, C.; Tian, G.; Meng, L.; Zhou, X. An improved gravitational search algorithm for profit-oriented par-tial disassembly line balancing problem. *Int. J. Prod. Res.* **2017**, *55*, 7302–7316. [CrossRef]
25. Kucukkoc, I. Balancing of two-sided disassembly lines: Problem definition, MILP model and genetic algorithm approach. *Comput. Oper. Res.* **2020**, *124*, 105064. [CrossRef]
26. Mete, S.; Çil, Z.A.; Özceylan, E.; Ağpak, K.; Battaïa, O. An optimisation support for the design of hybrid production lines including assembly and disassembly tasks. *Int. J. Prod. Res.* **2018**, *56*, 7375–7389. [CrossRef]
27. Zheng, F.; He, J.; Chu, F.; Liu, M. A new distribution-free model for disassembly line balancing problem with stochastic task processing times. *Int. J. Prod. Res.* **2018**, *56*, 7341–7353. [CrossRef]
28. He, J.; Chu, F.; Zheng, F.; Liu, M.; Chu, C. A multi-objective distribution-free model and method for stochastic disassembly line balancing problem. *Int. J. Prod. Res.* **2019**, *58*, 1–17. [CrossRef]
29. Zhang, Z.; Wang, K.; Zhu, L.; Wang, Y. A Pareto improved artificial fish swarm algorithm for solving a multi-objective fuzzy disassembly line balancing problem. *Expert Syst. Appl.* **2017**, *86*, 1339–1351. [CrossRef]
30. Kalayci, C.B.; Hancilar, A.; Gungor, A.; Gupta, S.M. Multi-objective fuzzy disassembly line balancing using a hybrid discrete artificial bee colony algorithm. *J. Manuf. Syst.* **2015**, *37*, 672–682. [CrossRef]
31. Chutima, P.; Chimklai, P. Multi-objective two-sided mixed-model assembly line balancing using particle swarm optimisation with negative knowledge. *Comput. Ind. Eng.* **2012**, *62*, 39–55. [CrossRef]
32. Lee, T.O.; Kim, Y.; Kim, Y.K. Two-sided assembly line balancing to maximize work relatedness and slackness. *Comput. Ind. Eng.* **2001**, *40*, 273–292. [CrossRef]
33. Tsaura, S.H.; Chang, T.Y.; Yen, C.H. The evaluation of airline service quality by fuzzy MCDM. *Tour. Manag.* **2002**, *23*, 107–115. [CrossRef]
34. Wang, K.; Li, X.; Gao, L.; Li, P. Energy consumption and profit-oriented disassembly line balancing for waste electrical and electronic equipment. *J. Clean. Prod.* **2020**, *265*, 121829. [CrossRef]
35. Li, D.; Zhang, C.; Tian, G.; Shao, X.; Li, Z. Multiobjective program and hybrid imperialist competitive algorithm for the mixed-model two-sided assembly lines subject to multiple constraints. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 119–129. [CrossRef]
36. Yuan, B.; Zhang, C.; Shao, X. A late acceptance hill-climbing algorithm for balancing two-sided assembly lines with multiple constraints. *J. Intell. Manuf.* **2013**, *26*, 159–168. [CrossRef]
37. Tseng, H.E.; Huang, Y.M.; Chang, C.C.; Lee, S.C. Disassembly sequence planning using a Flatworm algorithm. *J. Manuf. Syst.* **2020**, *57*, 416–428. [CrossRef]
38. Deb, K.; Agrawal, S.; Pratap, A.; Meyarivan, T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Lecture Notes in Computer Science*; Parallel Problem Solving from Nature PPSN VI, PPSN 2000; Schoenauer, M., Ed.; Springer: Berlin/Heidelberg, Germany, 2000; Volume 1917. [CrossRef]