

Article

Development of an Improved Water Cycle Algorithm for Solving an Energy-Efficient Disassembly-Line Balancing Problem

Xuesong Zhang ¹, Jing Yuan ¹, Xiaowen Chen ¹, Xingqin Zhang ¹, Changshu Zhan ^{1,*}, Amir M. Fathollahi-Fard ² , Chao Wang ³, Zhiming Liu ³ and Jie Wu ⁴

¹ Department of Automotive Service Engineering, School of Traffic and Transportation, Northeast Forestry University, Harbin 150040, China

² Department of Electrical Engineering, École de Technologie Supérieure, University of Québec, Montreal, QC H3C 1K3, Canada

³ Shandong Taizhan Electrom-Echanical Technology Co., Ltd., Zibo 255100, China

⁴ Qinghai Huasheng Ferroalloy Smelting Co., Ltd., Xining 810000, China

* Correspondence: zhchsh3@nefu.edu.cn; Tel.: +86-159-4599-2783

Abstract: Nowadays, there is a great deal of interest in the development of practical optimization models and intelligent solution algorithms for solving disassembly-line balancing problems. Based on the importance of energy efficiency of product disassembly and the trend for green remanufacturing, this paper develops a new optimization model for the energy-efficient disassembly-line balancing problem where the goal is to minimize the energy consumption generated during the disassembly-line operations. Since the proposed model is a complex optimization problem known as NP-hard, this study develops an improved metaheuristic algorithm based on the water cycle algorithm as a recently developed successful metaheuristic inspired by the natural water cycle phenomena of diversion, rainfall, confluence, and infiltration operations. A local search operator is added to the main algorithm to improve its performance. The proposed algorithm is validated by the exact solver and compared with other state-of-the-art and recent metaheuristic algorithms. A case study in a turbine reducer with different parameters is solved to show the applicability of this paper. Finally, our results confirm the high performance of the proposed improved water cycle algorithm and the efficiency of our sensitivity analyses during some sensitivity analyses.

Keywords: green remanufacturing; energy consumption; disassembly-line balancing problem; water cycle algorithm



Citation: Zhang, X.; Yuan, J.; Chen, X.; Zhang, X.; Zhan, C.; Fathollahi-Fard, A.M.; Wang, C.; Liu, Z.; Wu, J. Development of an Improved Water Cycle Algorithm for Solving an Energy-Efficient Disassembly-Line Balancing Problem. *Processes* **2022**, *10*, 1908. <https://doi.org/10.3390/pr10101908>

Academic Editor: Raul D.S.G. Campilho

Received: 21 August 2022

Accepted: 16 September 2022

Published: 21 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the development of the manufacturing industry and population growth, the number of retired products has increased leading to environmental pollution around the world [1]. The green remanufacturing concept using different recycling, disassembling, and remanufacturing activities for used and retired products is a solution to reduce the environmental pollution from the used products [2]. This concept is a motivation for this paper to work on Disassembly Sequence Planning (DSP) which is defined as a sequence of disassembly tasks to reduce the costs incurred during disassembly operations while increasing operational efficiency [3].

Since the DSP is not applicable to a large number of retired and end-of-life products to be scheduled simultaneously, recent studies shifted the DSP to the Disassembly-Line Balancing Problem (DLBP) to simplify disassembly activities while meeting various disassembly requirements [4–6]. The DLBP can better address the concept of green remanufacturing for balancing and improving the efficiency of remanufacturing disassembly operations [7–10].

Considering the impact of energy consumption on the DLBP [11–15], this paper is among the first studies to contribute to an energy-efficient DLBP where the goal is

to minimize the energy consumption generated during the disassembly-line operations. As a complex optimization problem known as NP-hard [16–36], the current existing algorithms may be inefficient for solving it, due to the no free lunch theory [37] for optimization problems. The main contribution of this paper is to propose an improved metaheuristic algorithm based on the water cycle algorithm as a recently developed successful metaheuristic inspired by the natural water cycle phenomena of diversion, rainfall, confluence, and infiltration operations. As far as we know from the literature, this metaheuristic algorithm has not been applied to solve any variations of DLBP before. In this paper, we have applied it and compared it with the state of the art and other recent metaheuristic algorithms.

The rest of this paper can be summarized as follows: Section 2 is a literature review of recent studies in the area of DLBP. Section 3 proposes the developed energy-efficient DLBP and its formulation. Section 4 introduces the improved water cycle algorithm using local search. Section 5 provides an extensive experimental study to analyze the performance of the proposed energy-efficient DLBP and our solution algorithm. Finally, a conclusion with findings, limitations, and recommendations from this paper is drawn in Section 6.

2. Literature Review

The research on the DLBP has evolved from building disassembly models to using intelligent algorithms employing different exact, heuristic, and metaheuristic algorithms. Henrioud et al. described the direct links between product structures in the association diagram model [3]. Li et al. proposed a hybrid diagram with which to express the geometric constraints of disassembled parts and the process of constraint dynamics' change, and this method can effectively simplify the process of generating disassembly sequences [4]. Lambert et al. proposed a graphical method based on AND/OR diagrams and applied it to the generation of optimal sequences [5]. Huang et al. proposed a disassembly matrix model using which meant that the effective sequence of disassembled units could be described quickly and accurately [6]. Mitrouchev et al. classified the disassembly hybrid diagrams by hierarchy to eliminate unrelated parts from the disassembly process and produce an optimal disassembly sequence [7]. Tian et al. introduced conflict matrices into AOG graphs to solve heterogeneous problems that cannot be solved by current heuristic demolition algorithms [8]. Wang et al. proposed a disassembly design method that combined regret theory with entropy weighting methods to provide systematic support for the evaluation of disassembly solutions [9]. Based on the Drosophila algorithm, Yuan et al. constructed a comprehensive evaluation model with multi-objective decomposition by introducing cross-efficiency and extended grey correlation. It was also optimally designed from three perspectives: time, economy, and environment, and finally an optimized solution was derived [10]. In another research paper, Smith and Chen proposed a rule-based approach to target selection disassembly [11].

An increase in the complexity of product construction is to have many parts involved in the disassembly, making it difficult to employ traditional algorithms for solving the DLBP [12]. This fact highlighted the development of intelligent heuristic and metaheuristic algorithms due to their fast solution speed, robustness, and ease of parameter adjustment, for example, in resource scheduling [13], vehicle path planning [14], system control [15], etc. Kongar and Gupta proposed an improved Genetic Algorithm (GA) for disassembly sequence planning [16]. Zhang et al. combined a disassembly hybrid graphical model with a Particle Swarm Optimization (PSO) algorithm for optimal disassembly sequence planning of complex products [17]. Xing et al. used Ant Colony Optimization (ACO) to find the feasible demolition sequences and find the relationship between the superiority and inferiority of each sequence to obtain the Pareto solution set [18]. Kongar and Gupta proposed an evolutionary algorithm to be used in the disassembly of used products [19]. Wang et al. used a combination of hybrid graphs and a GA to obtain the optimal disassembly sequence [20].

Recently, improved and modified metaheuristic algorithms have become an active research topic where many previous and recent algorithms have been combined with DLBP

problems to make improvements [21]. For example, Adenso-Diaz et al. proposed a greedy stochastic adaptive search algorithm based on path reconnection for the dual criteria disassembly problem [22]. Wu et al. proposed an improved genetic algorithm that improves on the traditional crossover and mutation process of genetic algorithms and applied it to the DLBP problem [23]. Kheder et al. considered factors such as machine maintainability, number of parts, number of tool changes, and changes in orientation during disassembly in the disassembly sequence planning process, and solved it using a GA [24]. Zhang et al. used the artificial bee colony (ABC) algorithm to solve a complex product disassembly sequence problem under parallel disassembly [25]. Ren proposed a genetic algorithm-based asynchronous parallel disassembly plan by adding operator constraints and tool number constraints to the disassembly sequence planning constraints [26]. Guo et al. proposed a dual-objective optimization model for maximizing disassembly profit and minimizing disassembly cost and solved their model using a decentralized search algorithm [27]. Ding et al. proposed a multi-objective ant colony algorithm based on Pareto dominance and applied it to 52 demolition task examples [28]. Zhang et al. improved the social engineering algorithm based on the DLBP problem by introducing the concept of swapping subsets and swapping sequences into the algorithm [29]. Tian et al. considered the fuzzy part quality and different operational-cost factor pairs in the DLBP problem and solved it using an ABC [30]. Yang et al. came out with a multi-objective disassembly line balancing the fruit fly optimization algorithm and applied it to the planning of disassembly sequences for outdated agricultural machinery [31]. In another study, Tian et al. proposed a fuzzy grey Choquet integral method for evaluating multi-criteria decision problems with interactive and qualitative indicators [32], optimizing the lambda fuzzy measure based on the weights given by the experts to enhance the consistency of the weights. Su et al. proposed a variable neighborhood search algorithm and designed a new local search process to solve the DLBP problem by introducing taboo tables into it [33]. Yang et al. proposed a target-selective disassembly sequence planning method that considered product failure characteristics to address the uncertainty and ambiguity of product quality in the actual disassembly process, especially the influence of the prevalent product failure problems on the selection of disassembly sequence solutions [34]. Bentaha et al. considered the uncertainty caused by product quality in the planning of the disassembly line and used product quality as an influencing factor in the allocation of disassembly tasks [35]. Last but not least, Tian et al. proposed a novel production and remanufacturing method for assessing the remanufacture of automotive parts, which combined fuzzy Analytic Hierarchy Process (AHP) and fuzzy G-topics [36].

Coming to a conclusion for this literature review, the first finding is that the energy-efficient DLBP is rarely formulated in the literature [12]. Another finding is that, although many metaheuristics, such as GA, PSO, ACO, and ABC, have been applied to different variations of DLBP, the water cycle algorithm (WCA), as a recently developed metaheuristic algorithm, has been never applied to this research area.

The WCA was proposed by Eskandar et al. [38], inspired by nature and from observing the flow of water from rivers and streams to the sea during the natural water cycle. The ocean is the best individual in the current population, the river is the next best individual, and the remaining inferior individuals are the streams. The WCA algorithm has been used in several applications, such as function optimization [39] and mechanical engineering optimization [40], and has often produced better results than other algorithms in these areas, demonstrating its good optimization capabilities.

As far as we know, there has been no research on the application of the WCA to DLBP problems. The WCA is simple and intuitive, easy to understand, and has a strong search capability. After testing the WCA on our energy-efficient DLBP, it can significantly improve disassembly efficiency and promote green production. Based on these reasons, an improved WCA is proposed in this paper, which defines the new encoding and decoding methods for DLBP problem characteristics. Three sink-update methods are introduced to improve the quality and diversity of the solution sequences. A local search method is introduced to improve the accuracy of the algorithm. Finally, the effectiveness and practicality of the

algorithm are investigated through an example of a worm reducer as our case study to verify the applicability of this paper.

3. Proposed Problem

To define the proposed problem, a disassembly mixture diagram was used to describe the information related to the parts to be disassembled and to describe the constraint relationships and hierarchical information between the parts [29]. The basic disassembly mixture diagram can be represented by $G = \{H, Z, S\}$, where $H = \{h_1, h_2, \dots, h_M\}$ denotes the parts of the product to be disassembled and M is the number of parts to be disassembled; $Z = \{z_1, z_2, \dots, z_l\}$ is the set of undirected edges of the product disassembly hybrid graph, indicating that the relationship between the two parts is in direct contact; $S = \{s_1, s_2, \dots, s_m\}$ is the set of directed edges of the product disassembly hybrid graph, indicating the existence of a priority relationship constraint between two disassembled parts. A basic disassembly hybrid graph is shown in Figure 1.

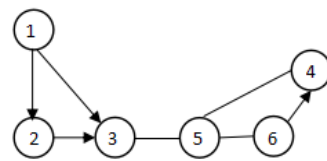


Figure 1. Disassembly Hybrid Graph.

The number in the circle indicates the task number to be disassembled; a solid line without an arrow indicates that the two parts to be disassembled are in direct contact with each other; a solid line with an arrow indicates that there is a forced disassembly priority between the two parts to be disassembled; the number at the beginning of the arrow is the task immediately preceding the number at the end.

Based on the set of vertices, the set of undirected edges and the set of directed edges of the hybrid graph model, the contact constraint matrix C and the priority constraint matrix P of the disassembled part can be structured:

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}$$

$C_{ij} = 1$ when there is direct contact between part i and part j . $C_{ij} = 0$ when they are not in direct contact or when $i = j$:

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}$$

$P_{ij} = 1$ when part i is the immediately preceding task of part j , otherwise $p_{ij} = 0$. When a part needs to be removed, it must satisfy the Equation (1) [29]:

$$\sum_{j=1}^n c_{ij} = 1 \quad \text{and} \quad \sum_{j=1}^n p_{ij} = 0 \quad (1)$$

Although many formulations have been developed to model the DLBP minimizing both disassembly cost and disassembly time [16–20], these models ignore the impact of energy consumption which is not negligible in DLBP problems. Based on this fact, this study proposes an energy-efficient DLBP which can effectively reduce the energy

consumption and resource waste generated in the disassembly process and enhance green remanufacturing. The notations in Table 1 are used in the proposed optimization model.

$$F = \sum_{n=1}^N \sum_{m=1}^M (1 + g_m) e_m t_m x_{mn} + e_w \sum_{n=1}^N \left(CT - \sum_{m=1}^M t_m x_{mn} \right) \quad (2)$$

Table 1. Symbols of equation parameters.

Indices:	
m :	Index of disassembly task numbers, $m = 1, 2, \dots, M$.
n :	Index of workstation numbers, $n = 1, 2, \dots, N$.
i/j :	Index of product part numbers.
Parameters:	
M :	Total number of disassembly tasks.
N :	Maximum number of workstations.
CT :	Cycle time of the disassembly line.
C :	Total number of components.
t_m :	Disassembly time for task m .
t_t :	Time required to change the disassembly tool.
t_d :	Time required to change the disassembly direction
e_m :	The unit time energy consumption of task m .
e_w :	The unit time energy consumption of workstation standby.
g_m :	Difficulty to remove a component in task m .
Decision variables:	
x_{mn} :	If task m is selected to be disassembled in workstation n , $x_{mn} = 1$, otherwise $x_{mn} = 0$.
y_m :	If the disassembly tool in task m is different from task $i-1$, $y_m = 1$, otherwise $y_m = 0$.
z_m :	If the disassembly direction in task m is different from task $i-1$, $z_m = 1$, otherwise $z_m = 0$.

Subject to:

$$\frac{\sum_{n=1}^N \sum_{m=1}^M (t_m x_{mn} + t_t y_m + t_d z_m)}{CT} \leq N \leq M \quad (3)$$

$$\sum_{m=1}^M t_m x_{mn} + t_t y_m + t_d z_m \leq CT, n = 1, 2, \dots, N \quad (4)$$

$$\sum_{n=1}^N x_{mn} = 1, \quad m = 1, 2, \dots, M \quad (5)$$

$$\sum_{j=1}^n c_{ij} = 1 \quad (6)$$

$$\sum_{j=1}^n p_{ij} = 0 \quad (7)$$

$$x_{mn}, y_m, z_m = \{0, 1\} \quad (8)$$

Equation (2) represents the objective function related to the energy consumption generated during the disassembly operation and the operation of the workstation. The energy consumption of the disassembly operation is multiplied by a factor considering that the more difficult the part to be disassembled, the more energy is generated for that task.

This objective function in Equation (2) is limited by the constraints of Equations (3) to (8). Equation (3) indicates that the number of workstations switched on does not exceed the maximum number of workstations and that the maximum number of workstations cannot exceed the number of disassembly tasks. Equation (4) indicates that the combined workstation dismantling tasks cannot exceed the workstation dismantling cycle time. Equation (5) indicates that each task can only be assigned to one workstation. Equations (6) and (7) represent the conditions to be met when the parts need to be disassembled. Finally, our binary decision variables are defined in relation to Equation (8).

4. Proposed Solution Method

4.1. Encoding and Decoding

The WCA works on a continuous search space, however, the search space for our optimization model is designed by integer numbers which are different alternatives of the disassembly sequence where our metaheuristic algorithm selects such disassembly sequences randomly and intelligently from this search space. To convert the continuous search space of WCA to the expression of the disassembly sequence, we need to apply the following encoding and decoding phases to represent our solutions.

In the encoding phase, a single layer of task-based encoding is used, which directly generates a sequence of real numbers $X_n = (1, 2, \dots, n)$ with the number of elements equal to the number of tasks to be disassembled [41]. Each member of an individual represents a disassembly task and is disassembled in turn to obtain a disassembly solution, e.g., an individual (2, 5, 7, 9, 8, 10, 11) represents a disassembly sequence as follows: 2-5-7-9-8-10-11.

The decoding process involves assigning feasible disassembly sequences to specific workstations to form a disassembly solution. It has the following three steps [41]:

Step 1: Enter the feasible disassembly sequence X , the cycle time CT , turn on the first workstation $AS = 1$, and the remaining allocable time at the current workstation is ST , so that $ST = CT$;

Step 2: In the order of part disassembly in sequence X , determine whether the operation time t_i of disassembly of parts i is greater than ST . If so, make $ST = CT$ and open a new workstation with $AS = AS + 1$, otherwise make $ST = ST - t_i$;

Step 3: Repeat Step 2 until task assignment for sequence X is complete and the disassembly solution is output.

4.2. Original WCA

The standard WCA is inspired by the flow of streams, rivers, and streams into the ocean during nature's water cycle [38]. The parameters of this algorithm are first initialized, after which the initial populations are randomly generated to form the initial streams, rivers, and oceans. The intensity of the raindrop flow to the river and the ocean is determined; the stream location is updated, and the river location is updated. If the stream gives a better fitness value than the river it is connected to, then the river and stream positions are swapped; if the river gives a better fitness value than the ocean it is connected to, then the ocean and river positions are swapped. If a river gives a better fitness value than its connected ocean, the ocean and river are swapped. If the evapotranspiration condition is met, then the precipitation process is initiated and new precipitation is formed. The process is then cycled through until the algorithm's termination condition is met and the optimal solution is output.

4.3. Improved WCA

Based on a study of the original WCA, this study adds new encoding and decoding methods, local search, and convergence update methods to redefine the steps of WCA as shown below.

Step 1: Initialize the disassembly sequence

As mentioned earlier, DLBP is a discrete optimization problem that needs to satisfy the disassembly relationship between the various parts [8]. Therefore, a feasible disassembly sequence needs to be generated before initialization. The solution sequence is generated based on the proposed encoding and decoding methods, and the proposed method for solution generation.

The sequence generation method is as follows: Firstly, two matrices are generated from the disassembly mixture diagram; the priority constraint matrix and the contact matrix. Second, the rows of the matrix that are eligible for disassembly are found, and one of the rows is randomly selected as the first component to be disassembled. The row is then deleted from the matrix and the above steps are repeated until the contact constraint matrix and the priority constraint matrix are empty, generating a feasible disassembly sequence.

Step 2: Determining the strength of the flow

The initial sea, rivers, and streams are determined after calculating the fitness values of each initial solution sequence, after which the number of rivers and the number of streams flowing to the sea and the number of streams flowing to the corresponding rivers in the current population are calculated according to Equations (9) and (10) [38]:

$$N_{stream} = N_{pop} - N_{sr} \quad (9)$$

$$N_{srn} = round \left(\left(\frac{Ct_n}{\sum_{s=1}^N Ct_s} \right) N_{streams} \right), n = 1, 2, \dots \quad (10)$$

where N_{pop} is the initial population number and N_{sr} is the number of sea and rivers.

Step 3: Convergence update process

This step discretizes the traditional water cycle sink update process by introducing three operations, one of which is chosen randomly during the sink update. Sequence diversity and sequence quality can be solved efficiently.

(1) When a stream or river flows towards the sea, a continuous part of the stream or river is selected, the disassembly sequence of the disassembled part corresponds to the part found in the sea, and the disassembly sequence in the stream is swapped according to the disassembly sequence in the sea. When the stream flows into the river, the disassembly sequence of the disassembled parts corresponding to the segment is found in the river, as shown in Figure 2.

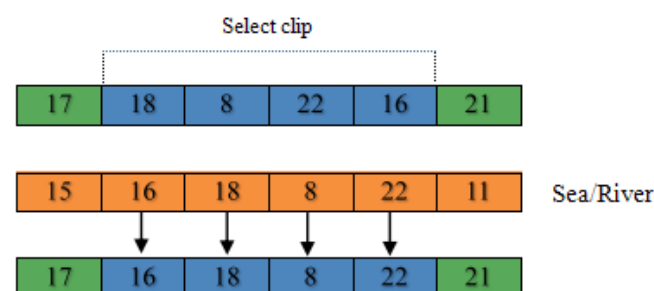


Figure 2. First type of confluence update.

(2) Randomly select two disassembly tasks in the solution sequence and invert the sequence of the two tasks and the disassembly tasks between them to generate a new sequence, as shown in Figure 3.

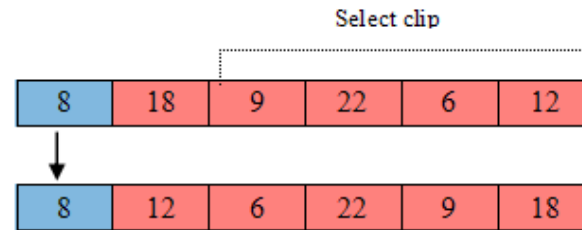


Figure 3. Second type of confluence update.

(3) When a stream or river flows towards the sea, a part of the continuous segment in the sea is selected, the same position sequence in the stream or river is replaced with a new continuous segment, and the remaining sequences are rearranged according to priority relations. In the case of a stream flowing into a river, a portion of the continuous segments in the river is selected, as shown in Figure 4.

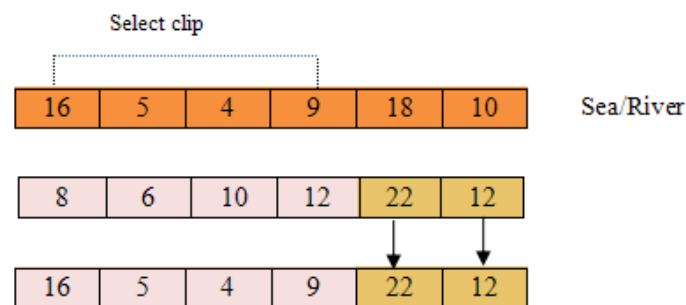


Figure 4. Third type of confluence update.

Step 4: Evaporation and rainfall process using the local search

To improve the accuracy of the algorithm and prevent local convergence, the traditional processes of evaporation and rainfall in the water cycle are changed into a local search process. After the sink update is completed, for any solution sequence, a random number is generated, and if the random number is smaller than a given value, then a local search is performed, and if the new solution has a higher fitness than the original solution, then the original solution is replaced. The local search operation is a random exchange of three randomly selected points in the solution sequence, as shown in Figure 5.

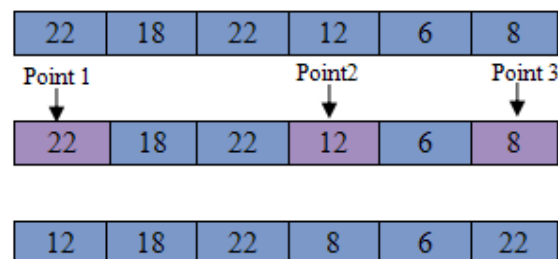


Figure 5. Local search method.

Step 5: Correction sequence

Starting from the first position of the generated disassembly sequence, if the part represented by this position is disassembled, check the disassembly capability of the

next part. If it does not satisfy the disassembly requirements, replace it with a randomly selected part that can be disassembled and check that the next part meets the disassembly requirements. Repeat until the last part meets the disassembly requirements.

Finally, the flowchart of the proposed algorithm is shown in Figure 6.

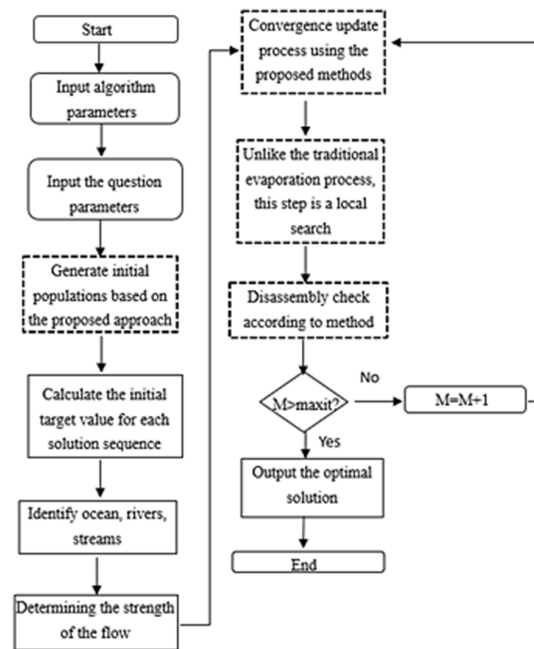


Figure 6. WCA algorithm flow chart.

5. Discussion and Results

Here, we first analyze the optimization model and solution algorithm in calibrated datasets and then random ones. To run the tests and algorithms, all of the codes were written in MATLAB software (<https://www.mathworks.com/products/matlab.html> accessed on 8 January 2021) on an operating system using Intel(R) Core (TM) i7-10850H CPU @ 2.70GHz, 2712 Mhz, 6 Core(s) and 12 Logical Processor(s).

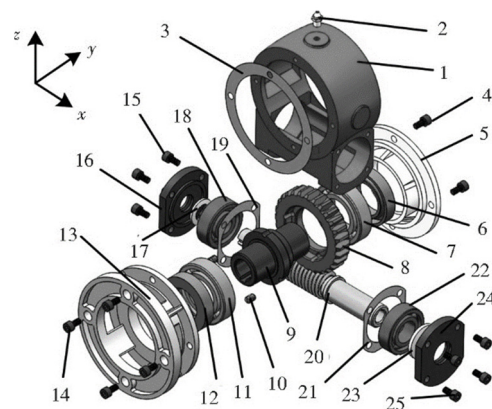


Figure 7. Drawing of worm reducer.

5.1. Calibrated Data Sets

5.1.1. A Case Study for Our Model

In order to verify the effectiveness of the proposed algorithm, a worm reducer with 25 components from the literature [34] is solved, as shown in Figure 7. According to the product assembly relationship and spatial location constraints, a disassembly hybrid dia-

gram of the turbo reducer can be obtained, as shown in Figure 8. The original disassembly information of each part is shown in Table 2.

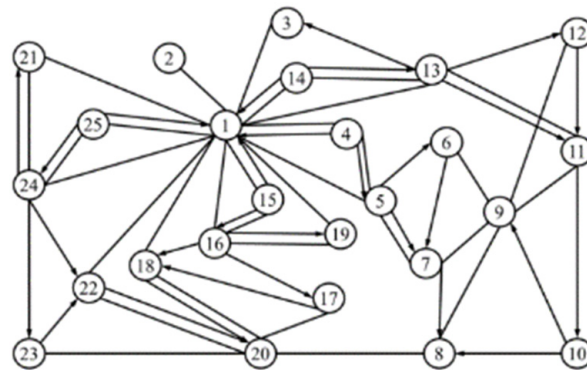


Figure 8. Hybrid diagram of worm reducer disassembly.

Table 2. Parts of turbine reducer.

Order	Name	Quantity	Tool	Task Difficulty	Disassembly Time/s	Direction
1	Shell (non-removable)	1	-	-	-	
2	Grease fitting	1	Wrench (T ₁)	0.2	18	+z
3	Turbine shaft shim end cover	1	Special tool (T ₂)	1.2	5	-y
4	Hexagon socket head cap screws	4	Allen wrench (T ₃)	0	25	+y
5	Turbine shaft end cover 1	1	Hand (T ₀)	1	10	+y
6	Skeleton oil seal 1	1	Hammer (T ₄)	1	8	+y
7	Turbine shaft bearing 1	1	Hammer (T ₄)	1	15	+y
8	Turbine	1	Special tool (T ₅)	1	8	+y
9	Turbine shaft	1	Hammer (T ₄)	1	8	-y
10	Slotted set screws with flat point	3	Screwdriver (T ₆)	0	30	-y
11	Turbine shaft bearing 2	1	Hammer (T ₄)	1	15	-y
12	Skeleton oil seal 2	1	Hammer (T ₄)	0.8	8	-y
13	Turbine shaft end cover 2	1	Hand (T ₀)	0.2	10	-y
14	Hexagon socket head cap screws	4	Allen wrench (T ₃)	0	25	-y
15	Hexagon socket head cap screws	4	Allen wrench (T ₃)	0	25	-x
16	Worm shaft end cover 1	1	Hand (T ₀)	1	8	-x
17	Oil seal 1	1	Tong (T ₇)	1	6	-x
18	Worm shaft bearing 1	1	Hammer (T ₄)	1	15	-x
19	Bearing cap gasket 1	1	Special tool (T ₂)	1	5	-x
20	Worm	1	Special tool (T ₅)	1	8	-x
21	Bearing cap gasket 2	1	Special tool (T ₂)	0.4	5	+x
22	Worm shaft bearing 2	1	Hammer (T ₄)	0.4	15	+x
23	Oil seal 2	1	Tong (T ₇)	1	6	+x
24	Worm shaft end cover 2	1	Hand (T ₀)	0.2	8	+x
25	Hexagon socket head cap screws	4	Allen wrench (T ₃)	0	25	+x

It should be noted that we have set $e_m = 0.8$, $e_w = 0.2$, $t_t = 4s$, $t_d = 8s$, and $CT = 120$, combined with the information in Table 1. The line-balancing solutions for worm reducer disassembly lines can be performed by following the methodological steps developed.

5.1.2. Analysis of Algorithm's Parameters

According to Section 4, Nrs is the parameter that influences the improved WCA. To analyze the sensitivity of the parameters in the WCA, three different Nrs values were substituted into the procedure. The number of iterations was 200, initial population of 20. The program was run fifteen times, and the final results are shown below. The circles in the figures represent outliers.

From Figures 9 and 10, it can be seen that as Nrs expands, the higher the accuracy of the solution result and the closer to the optimal solution, but as Nrs increases, the solution

time will also increase. The balance between time and solution accuracy is what we need to consider.

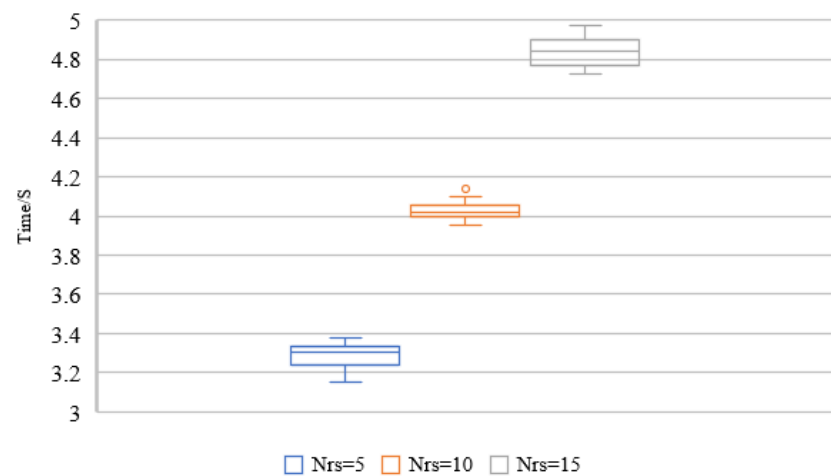


Figure 9. Box plots of time indicators.

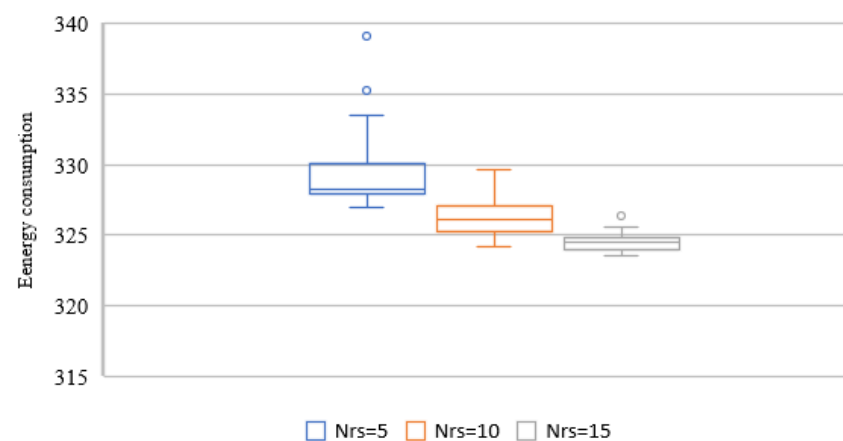


Figure 10. Box plot of objective function values.

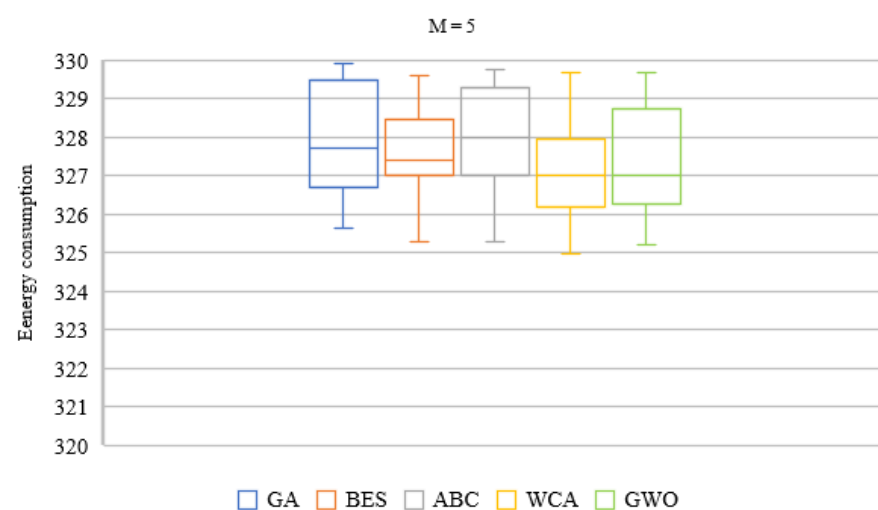


Figure 11. Box plot of the objective function at an initial population of 5.

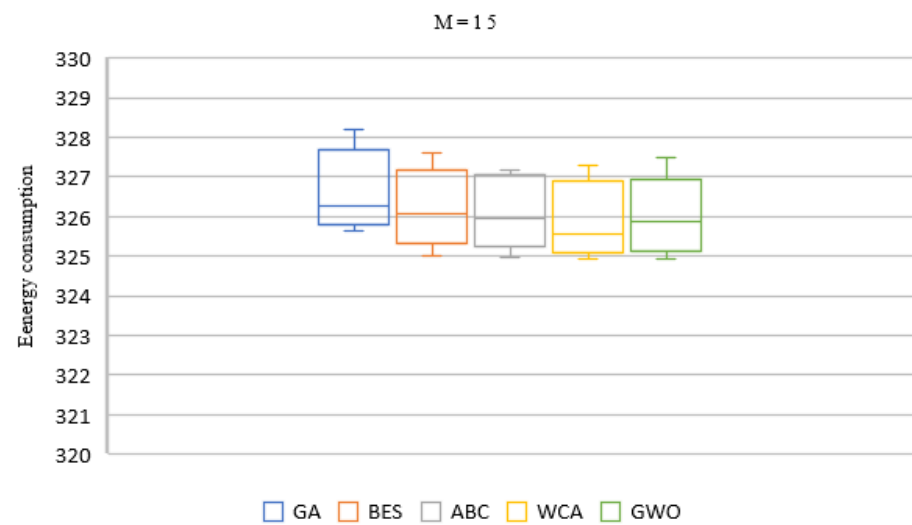


Figure 12. Box plot of the objective function at an initial population of 15.

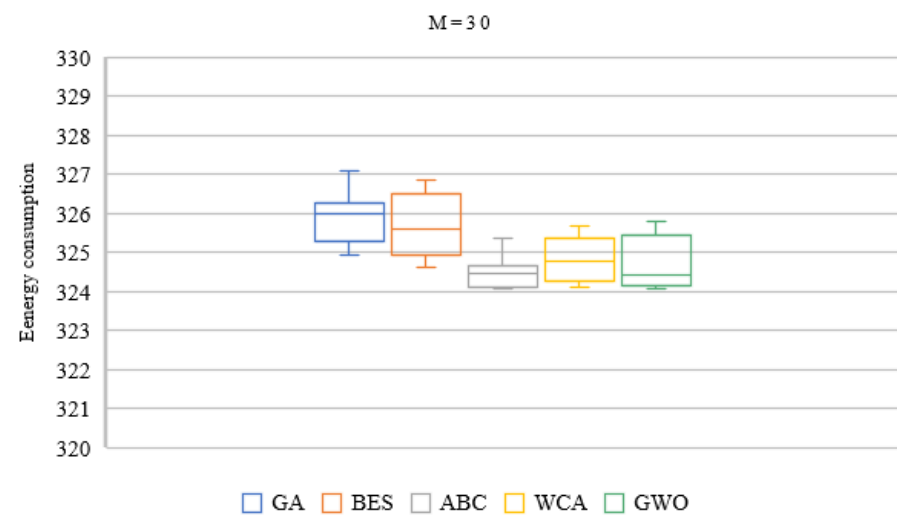


Figure 13. Box plot of the objective function at an initial population of 30.

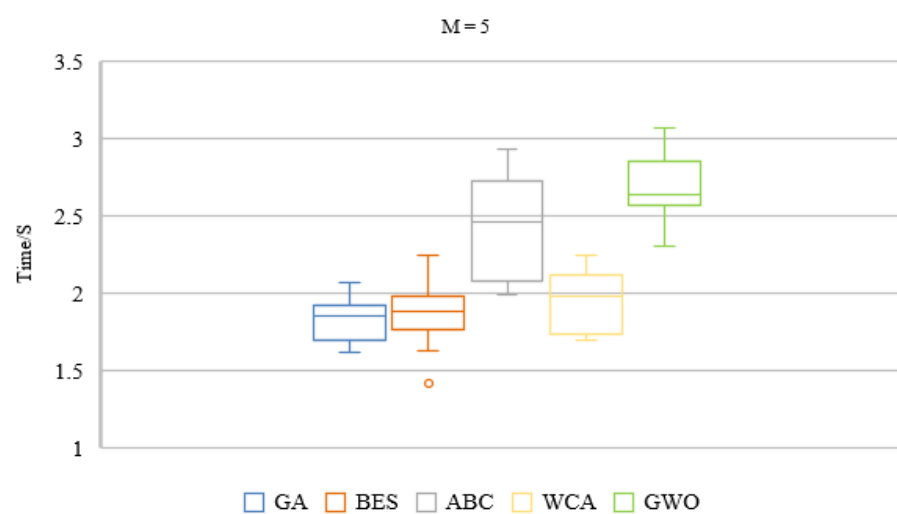


Figure 14. Running time box plot when the population is 5.

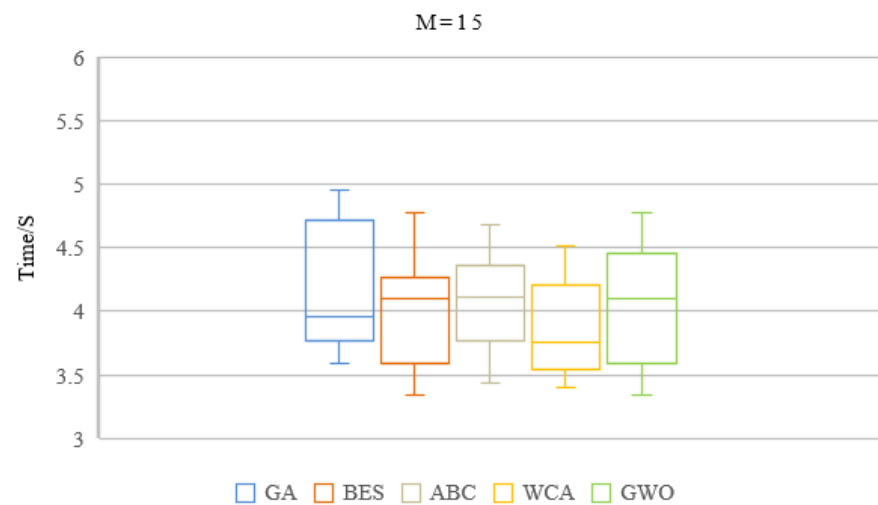


Figure 15. Running time box plot when the population is 15.

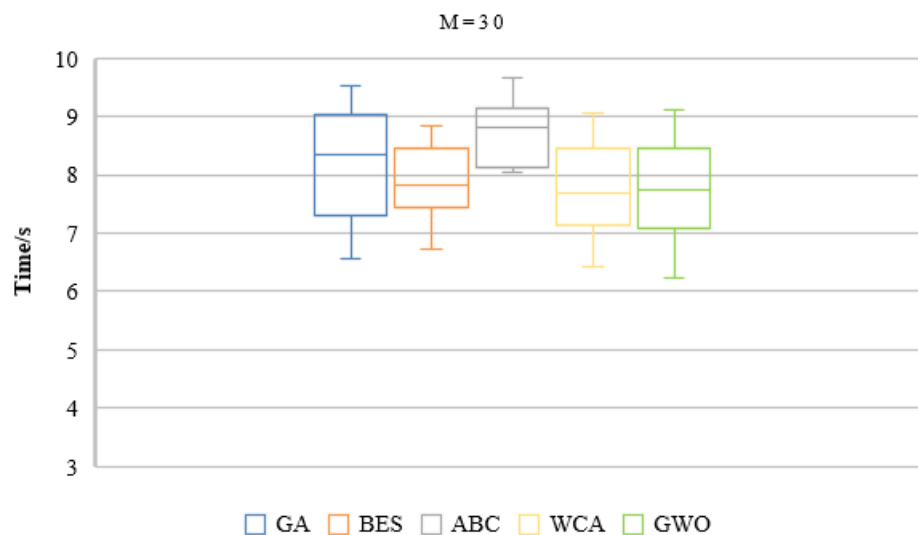


Figure 16. Running time box plot when the population is 30.

5.1.3. Comparison with Other Algorithms

To verify the effectiveness of the WCA, the disassembly sequence of the worm reducer was solved using the GA, bald eagle search (BES), ABC, grey wolf optimizer (GWO), and our improved WCA. To have a fair comparison, the number of iterations and populations in these algorithms are set as the same. In this regard, the initial population sizes were 5, 15, and 30, the number of iterations was 200, Nrs were 3, 10, and 18, and the other parameters were the same as in the previous section. Since all of the metaheuristic works randomly, each algorithm was run 15 times to obtain reliable results. In this regard, we have completed some statistical tests, as shown in Figures 11–16, where box-line plots for these five algorithms in different numbers of initial population sizes are analyzed.

Figures 11–16 show that ABC gives the best and most stable optimization results when the population size reaches a certain number. However, its computational run time is longer. When the population size is larger, the computational running time of WCA and GWO is in an advantageous position. However, the quality of WCA's optimization results is better than the ones obtained by GWO. When the population size is lower, the solution results of GA, ABC, and GWO are more unstable, and the solution quality is not as good as that of WCA, which can maintain better accuracy and better program transportation time.

As can be seen in Figure 17, WCA has good convergence and solution quality. In addition, WCA maintains a good balance between solution quality, solution stability, and solution efficiency and is superior. Therefore, it can be better adapted to the actual dismantling line-operation process where the situation is variable and can effectively improve the disassembling efficiency.

To analyze the convergence behavior, the initial population size was set to 50, Nrs to 26, the number of iterations to 200, and other parameters were the same as in the previous section. The convergence curves of each algorithm are shown in Figure 17, and the line balancing scheme is shown in Table 3.

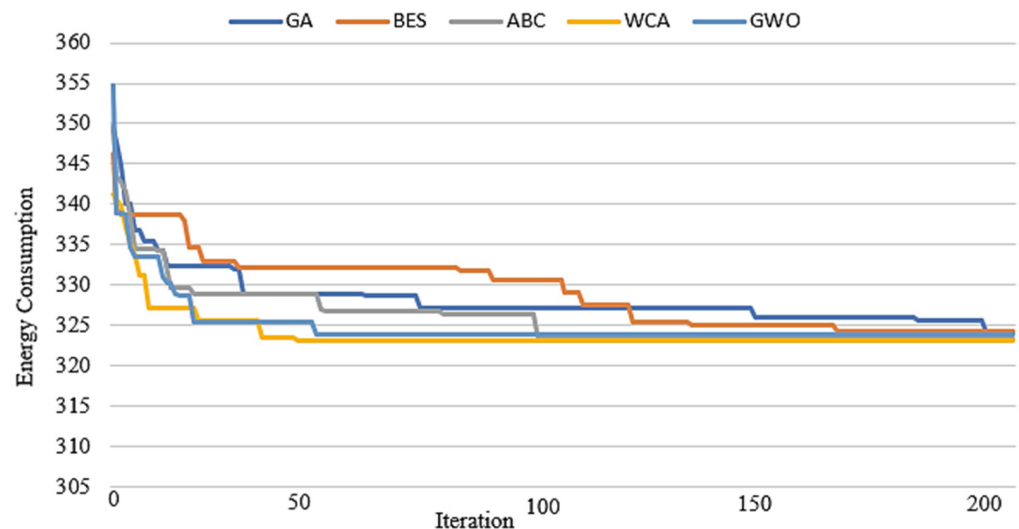


Figure 17. Convergence curves for the five algorithms.

Table 3. Five algorithms for disassembly sequences.

Order	Line Balance Scheme	<i>F</i>
GA	[25, 2, 24, 15] → [14, 16, 21, 13, 17] → [4, 12, 19, 11, 3,] → [18, 5, 23, 10, 6] → [7, 9, 28, 22, 20]	324.04
BES	[15, 2, 16, 14] → [25, 13, 3, 4, 12] → [11, 5, 17, 6, 24] → [10, 18, 21, 7, 23] → [9, 8, 19, 22, 20]	324.2
ABC	[15, 2, 16, 4] → [25, 5, 17, 14] → [18, 24, 19, 13, 23, 12] → [11, 6, 3, 7, 21] → [10, 8, 9, 22, 20]	323.24
WCA	[25, 2, 24, 4] → [15, 5, 21, 14, 23] → [13, 6, 3, 7, 16, 12] → [22, 11, 17, 10] → [18, 8, 19, 9, 20]	323.08
GWO	[25, 2, 24, 4] → [15, 5, 23, 14] → [22, 13, 6, 16, 12, 17] → [7, 18, 21, 11, 19] → [10, 8, 3, 20, 9]	323.88

5.2. Random Data Sets

5.2.1. Data Generation

To further analyze the performance of the algorithm, the proposed algorithm is applied to 19 random case instances with different task sizes. The problem data were generated in such a way that the total number of disassembly tasks was 8:4:80 (the total number of tasks for the first task was 8; each subsequent task was increased by 4; the total number of disassembly tasks was incremented to 80). Task 1 has a disassembly time of 3 s, task 2 has a disassembly time of 5 s, task 3 has a disassembly time of 7 s, task 4 has a disassembly time of 11 s, and so on, defining the last task with a disassembly time of 11 s corresponding to a part/component that is dangerous, the last task with a disassembly time of 7 s corresponding to a part/component requirement of 1, and a production beat of 26 s. The theoretical

optimal solution to the test problem is $f_1 = n/4$, $f_2 = 0$, $f_3 = 1$, $f_4 = 2$, f_1 is the number of workstations turned on, f_2 is the target smoothing index, f_3 is the target hazard index and f_4 is the target demand index. Defining a fixed demand index of 1. It should be noted that in our metaheuristic algorithm, the initial population size was set to 50, Nrs to 26, the number of iterations to 50, and the other parameters are shown in the previous table.

5.2.2. Solving Random Tests by Our Improved WCA

The experiment was tested three times for each problem and the average was taken where their results are reported in Table 4 for the proposed WCA.

Table 4. Results of random tests.

Number of Parts	f_1	f_2	f_3	f_4
8	2	0	1	2
12	3	0	1	2
16	4	0	1	2
20	5	0	1	2
24	6	0	1	2
28	7	0	1	2
32	8	0	1	2
36	9	0	1	2
40	10	0	1	2
44	11	0	1	2
48	12	0	1	2
52	13	0	1	2
56	14	0	1	2
60	15	0	1	2
64	17	50	1	2
68	18	44	1	2
72	19	42	1	2
76	20	38	1	3
80	21	36	1	3

As can be seen from Table 4, the algorithm proposed in this paper achieves optimal solutions for 14 problems in terms of the target workstation number f_1 and the target smoothing index f_2 , and for all problems in terms of the target hazard index f_3 , 17 problems in terms of target demand index f_4 . The algorithm has good solution quality for small-, medium-, and large-scale cases.

6. Conclusions and Future Works

In this paper, we proposed an improved WCA that enables it to solve discrete optimization problems for the application of an energy-efficient DLBP. In our proposed method, the encoding and decoding methods and three sequential update operations were designed, and a local search strategy was introduced to improve the diversity and quality of the solution set, which can quickly identify near-optimal disassembly solutions. At the same time, we constructed a disassembly-line adaptation function with energy consumption as the target, where the disassembly solution obtained by our WCA metaheuristic can effectively reduce the energy consumption generated during the operation of the disassembly line, improve the efficiency and effectiveness of disassembly, and promote the green and sustainable development of disassembly. The solution of the operation example using the worm reducer validates the applicability of our proposed algorithm. Finally, the main finding from our results is that the improved algorithm can be more effective and efficient than existing algorithms.

Although this paper analyzed an energy-efficient DLBP using an improved WCA, there are many research directions for future work. For example, there are many uncertainties in the actual operation of the disassembling line, such as faulty parts [42], worker factors [43], fuzzy working time [44], environmental factors [45,46], etc. The introduction of multi-criteria decision making [47], combined with stochastic simulation methods [48], and new information technologies in DLBP is one potential future research direction [29]. Finally, the

line balancing considerations for multiple batches and types of products with ambiguous dismantling times is another area that can be investigated in the future.

Author Contributions: Conceptualization, Methodology and Software, C.Z.; Data Collection, Writing and Original Draft Preparation, X.Z. (Xuesong Zhang); Visualization and Investigation, J.Y. and X.C.; Supervision, X.Z. (Xingqin Zhang). and A.M.F.-F.; Software and Validation, C.W., J.W. and Z.L.; Review and editing, A.M.F.-F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Tian, G.; Yuan, G.; Aleksandrov, A.; Zhang, T.; Li, Z.; Fathollahi-Fard, A.M.; Ivanov, M. Recycling of spent Lithium-ion Batteries: A comprehensive review for identification of main challenges and future research trends. *Sustain. Energy. Techn.* **2022**, *53*, 102447. [\[CrossRef\]](#)
2. Guo, L.; Zhang, X. Remanufacturing parallel disassembly sequence planning method driven by multiple failures. *J. Zhejiang Univ. Eng. Sci.* **2020**, *54*, 2233–2246.
3. Henrioud, J.M.; Bourjault, A.L. *A Computer-Aided Generator of Assembly Plans*; Springer: Boston, MA, USA, 1991; pp. 191–215.
4. Li, J.; Khoo, L.P.; Tor, S.B. A novel representation scheme for disassembly sequence planning. *Int. J. Adv. Manuf. Tech.* **2002**, *20*, 621–630. [\[CrossRef\]](#)
5. Lambert, A.J.D. Optimal disassembly of complex products. *Int. J. Prod. Res.* **1997**, *35*, 2509–2523. [\[CrossRef\]](#)
6. Huang, Y.; Huang, C. Disassembly matrix for disassembly processes of products. *Int. J. Prod. Res.* **2002**, *40*, 255–273. [\[CrossRef\]](#)
7. Mitrouchev, P.; Wang, C.; Lu, L.; Li, G. Selective disassembly sequence generation based on lowest level disassembly graph method. *Int. J. Adv. Manuf. Tech.* **2015**, *80*, 141–159. [\[CrossRef\]](#)
8. Tian, G.; Ren, Y.; Feng, Y.; Zhou, M.; Zhang, H.; Tan, J. Modeling and planning for dual-objective selective disassembly using AND/OR graph and discrete artificial bee colony. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2456–2468. [\[CrossRef\]](#)
9. Wang, W.; Tian, G.; Zhang, T.; Jabarullah, N.H.; Li, F.; Fathollahi-Fard, A.M.; Wang, D.; Li, Z. Scheme selection of design for disassembly (DFD) based on sustainability: A novel hybrid of interval 2-tuple linguistic intuitionistic fuzzy numbers and regret theory. *J. Clean. Prod.* **2020**, *281*, 124724. [\[CrossRef\]](#)
10. Yuan, G.; Yang, Y.; Tian, G.; Zhuang, Q. Comprehensive evaluation of disassembly performance based on the ultimate cross-efficiency and extension-gray correlation degree. *J. Clean. Prod.* **2020**, *245*, 118800. [\[CrossRef\]](#)
11. Smith, S.S.; Chen, W. Rule-based recursive selective disassembly sequence planning for green design. *Adv. Eng. Inform.* **2011**, *25*, 77–87. [\[CrossRef\]](#)
12. Tian, G.; Zhang, C.; Fathollahi-Fard, A.M.; Li, Z.; Zhang, C.; Jiang, Z. An Enhanced Social Engineering Optimizer for Solving an Energy-Efficient Disassembly Line Balancing Problem Based on Bucket Brigades and Cloud Theory. *IEEE Trans. Ind. Inform.* **2022**. [\[CrossRef\]](#)
13. Tian, G.; Fathollahi-Fard, A.M.; Ren, Y.; Li, Z.; Jiang, X. Multi-objective scheduling of priority-based rescue vehicles to extinguish forest fires using a multi-objective discrete gravitational search algorithm. *Inform. Sci.* **2022**, *608*, 578–596. [\[CrossRef\]](#)
14. Gholizadeh, H.; Fazlollahtabar, H.; Fathollahi-Fard, A.M.; Dulebenets, M. Preventive maintenance for the flexible flowshop scheduling under uncertainty: A waste-to-energy system. *Environ. Sci. Pollut. R.* **2021**, 1–20. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Fathollahi-Fard, A.M.; Woodward, L.; Akhrif, O. Sustainable distributed permutation flow-shop scheduling model based on a triple bottom line concept. *J. Ind. Inf. Integr.* **2021**, *24*, 100233. [\[CrossRef\]](#)
16. Kongar, E.; Gupta, S.M. Disassembly sequencing using genetic algorithm. *Int. J. Adv. Manuf. Tech.* **2006**, *30*, 497–506. [\[CrossRef\]](#)
17. Zhang, X.; Zhang, S. Product disassembly sequence planning based on particle swarm optimization algorithm. *Comput. Integr. Manuf. Syst.* **2009**, *15*, 508–514.
18. Xing, Y.; Wang, C.; Liu, Q. Disassembly sequence planning based on pareto ant colony algorithm. *J. Mech. Eng.* **2012**, *48*, 186–192. [\[CrossRef\]](#)
19. Gungor, A.; Gupta, S.M. Disassembly sequence planning for products with defective parts in product recovery. *Comput. Ind. Eng.* **1998**, *35*, 161–164. [\[CrossRef\]](#)
20. Wang, J.; Liu, H.; Li, S.; Zhong, Y. Intelligent selective disassembly using the ant colony algorithm. *Artifi. Intell. Eng. Des. Anal. Manuf.* **2003**, *17*, 325–333. [\[CrossRef\]](#)
21. Zhang, C.; Fathollahi-Fard, A.M.; Li, J.; Tian, G.; Zhang, T. Disassembly sequence planning for intelligent manufacturing using social engineering optimizer. *Symmetry* **2021**, *13*, 663. [\[CrossRef\]](#)
22. Adenso-Diaz, B.; Garcia-Carbajal, S.; Gupta, S.M. A path-relinking approach for a bi-criteria disassembly sequencing problem. *Comput. Oper. Res.* **2008**, *35*, 3989–3997. [\[CrossRef\]](#)

23. Wu, H.; Zuo, H. Product disassembly sequence planning based on improved genetic algorithm. *China Mech. Eng.* **2009**, *20*, 699–703.
24. Kheder, M.; Trigui, M.; Aifaoui, N. Disassembly sequence planning based on a genetic algorithm. *Proc. Inst. Mech. Eng. C J. Mech. Eng. Sci.* **2015**, *229*, 2281–2290. [[CrossRef](#)]
25. Zhang, L.; Peng, H.; Bian, B. Parallel disassembly modeling and planning method of complex products. *Chn. Mech. Eng.* **2014**, *25*, 937–943. [[CrossRef](#)]
26. Ren, Y.; Tian, G.; Zhao, F.; Yu, D.; Zhang, C. Selective cooperative disassembly planning based on multiobjective discrete artificial bee colony algorithm. *Eng. Appl. Artif. Intell.* **2017**, *64*, 415–431. [[CrossRef](#)]
27. Guo, X.; Liu, S.; Zhou, M.; Tian, G. Dual-objective program and scatter search for the optimization of disassembly sequences subject to multiresource constraints. *IEEE Trans. Autom. Sci. Eng.* **2018**, *15*, 1098–1103. [[CrossRef](#)]
28. Ding, L.; Tan, J.; Feng, Y. Multi-objective optimization of disassembly line balance based on Pareto ant colony algorithm. *Comput. Integr. Manu. Syst.* **2009**, *15*, 1406–1413.
29. Tian, G.; Zhang, Z.; Zhang, C.; Zhang, L. Multi-objective optimal design of disassembly line balance based on improved social engineering algorithm. *Packag. Eng.* **2022**, *43*, 7.
30. Tian, G.; Zhou, M.; Li, P. Disassembly sequence planning considering fuzzy component quality and varying operational cost. *IEEE Trans. Autom. Sci. Eng.* **2018**, *15*, 748–760. [[CrossRef](#)]
31. Yang, Y.; Yuan, G.; Zhuang, Q.; Tian, G. Multi-objective low-carbon disassembly line balancing for agricultural machinery using MDFOA and fuzzy AHP. *J. Clean. Prod.* **2019**, *233*, 1465–1474. [[CrossRef](#)]
32. Tian, G.; Hao, N.; Zhou, M.; Pedrycz, W.; Zhang, C.; Ma, F.; Li, Z. Fuzzy grey Choquet integral for evaluation of multicriteria decision making problems with interactive and qualitative indices. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *5*, 1855–1868. [[CrossRef](#)]
33. Su, Y.; Zhang, Z.; Hu, Y. A variable neighborhood search algorithm for solving demolition line equilibrium problems. *Mod. Manu. Eng.* **2016**, *10*, 19–25.
34. Yang, D.; Xu, Z.; Zhu, Z.; Su, K.; Liu, W. Target-selective demolition sequence planning considering product fault characteristics. *J. Harbin. Inst. Tech.* **2019**, *7*, 160–170.
35. Bentaha, M.L.; Voisin, A.; Marangé, P. A decision tool for disassembly process planning under end-of-life product quality. *Int. J. Prod. Econ.* **2020**, *219*, 386–401. [[CrossRef](#)]
36. Tian, G.; Zhang, H.; Feng, Y.; Jia, H.; Zhang, Y.; Jiang, G.; Li, W.; Li, P.G. Operation patterns analysis of automotive components remanufacturing industry development in China. *J. Clean. Prod.* **2017**, *64*, 1363–1375. [[CrossRef](#)]
37. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evolut. Comput.* **1997**, *1*, 67–82. [[CrossRef](#)]
38. Eskandar, H.; Sadollah, A.; Bahreininejad, A.; Hamdi, M. Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* **2012**, *110*, 151–166. [[CrossRef](#)]
39. Nasir, M.; Sadollah, A.; Choi, Y.H.; Kim, J.H. A comprehensive review on water cycle algorithm and its applications. *Neural. Comput. Appl.* **2020**, *32*, 17433–17488. [[CrossRef](#)]
40. Haddad, O.B.; Moravej, M.; Loáiciga, H.A. Application of the water cycle algorithm to the optimal operation of reservoir systems. *J. Irrig. Drain. Easce* **2015**, *141*, 04014064. [[CrossRef](#)]
41. Zhang, Q.; Cai, N.; Zeng, Y.; Li, L.; Zou, B. A review of modeling theory and solution methods for remanufacturing-oriented disassembly line balancing problems. *Chn. Mech. Eng.* **2018**, *21*, 2636–2645.
42. Ke, H.; Liu, H.; Tian, G. An Uncertain Random Programming Model for Project Scheduling Problem. *Int. J. Intell. Syst.* **2015**, *30*, 66–79. [[CrossRef](#)]
43. Tian, G.; Liu, Y.; Ke, H.; Chu, J. Energy evaluation method and its optimization models for process planning with stochastic characteristics: A case study in disassembly decision-making. *Comput. Ind. Eng.* **2012**, *63*, 553–563. [[CrossRef](#)]
44. Tian, G.; Chu, J.; Liu, Y.; Ke, H.; Zhao, X.; Xu, G. Expected energy analysis for industrial process planning problem with fuzzy time parameters. *Comput. Chem. Eng.* **2011**, *35*, 2905–2912. [[CrossRef](#)]
45. Xu, G. Precision Evaluation of Three-dimensional Feature Points Measurement by Binocular Vision. *J. Opt. Soc. Korea* **2011**, *31*, 30–37. [[CrossRef](#)]
46. Zhang, H.; Peng, Y.; Tian, G.; Wang, D.; Xie, P. Green material selection for sustainability: A hybrid MCDM approach. *PLoS ONE* **2017**, *12*, e0177578. [[CrossRef](#)] [[PubMed](#)]
47. Feng, Y.; Zhang, Z.; Tian, G.; Fathollahi-Fard, A.M.; Hao, N.; Li, Z.; Tan, J. A novel hybrid fuzzy grey TOPSIS method: Supplier evaluation of a collaborative manufacturing enterprise. *Appl. Sci.* **2019**, *9*, 3770. [[CrossRef](#)]
48. Tian, G.; Zhou, M.; Chu, J.; Qiang, T.; Hu, H. Stochastic cost-profit tradeoff model for locating an automotive service enterprise. *IEEE Trans. Autom. Sci. Eng.* **2014**, *12*, 580–587. [[CrossRef](#)]