automation

MDPI

Article

Solving a Stochastic Multi-Objective Sequence Dependence Disassembly Sequence Planning Problem with an Innovative Bees Algorithm

Xinyue Huang, Xuesong Zhang, Yanlong Gao and Changshu Zhan *

Mechanical and Electrical Engineering College, Northeast Forestry University, Harbin 150040, China; h1220xy@nefu.edu.cn (X.H.); xuesongzhang@nefu.edu.cn (X.Z.); gaoyanlong@nefu.edu.cn (Y.G.)
* Correspondence: zhchsh3@nefu.edu.cn

Abstract: As the number of end-of-life products multiplies, the issue of their efficient disassembly has become a critical problem that urgently needs addressing. The field of disassembly sequence planning has consequently attracted considerable attention. In the actual disassembly process, the complex structures of end-of-life products can lead to significant delays due to the interference between different tasks. Overlooking this can result in inefficiencies and a waste of resources. Therefore, it is particularly important to study the sequence-dependent disassembly sequence planning problem. Additionally, disassembly activities are inherently fraught with uncertainties, and neglecting these can further impact the effectiveness of disassembly. This study is the first to analyze the sequence-dependent disassembly sequence planning problem in an uncertain environment. It utilizes a stochastic programming approach to address these uncertainties. Furthermore, a mixed-integer optimization model is constructed to minimize the disassembly time and energy consumption simultaneously. Recognizing the complexity of the problem, this study introduces an innovative bees algorithm, which has proven its effectiveness by showing a superior performance compared to other state-of-the-art algorithms in various test cases. This research offers innovative solutions for the efficient disassembly of end-of-life products and holds significant implications for advancing sustainable development and the recycling of resources.

Keywords: sequence-dependent disassembly sequence planning; uncertain; stochastic programming; bees algorithm

Citation: Huang, X.; Zhang, X.; Gao, Y.; Zhan, C. Solving a Stochastic Multi-Objective Sequence Dependence Disassembly Sequence Planning Problem with an Innovative Bees Algorithm. *Automation* **2024**, *5*, 432–449. https://doi.org/10.3390/automation5030025

Academic Editors: Zude Zhou, Quan Liu, Wenjun Xu, F. Javier Ramírez, Marcello Fera, Mario Caterino, Duc Truong Pham and Jeremy Rickli

Received: 9 July 2024
Revised: 2 August 2024
Accepted: 15 August 2024
Published: 23 August 2024

1. Introduction

As our awareness of resource recycling and reuse grows, the effective disassembly of end-of-life (EOL) products has become a point of focus in both industry and academia [1]. Disassembly sequence planning (DSP) is an optimization approach designed to determine the most efficient sequence for a product's disassembly, which can significantly enhance the efficiency of the disassembly process, reduce the disassembly cycle time, and lower the associated costs [2].

The DSP problem has garnered considerable interest in recent years. Ren et al. introduced an AND/OR graph (AOG) to delineate disassembly sequences, offering a structured approach to the problem [3]. Li et al. advanced the field by proposing a novel representation method that captures both geometric and priority constraints during product disassembly. Their process was compared against the AOG and Petri nets, effectively showcasing its superiority [4]. Wang et al. presented the disassembly feasibility information graph, a model that reformulates the DSP problem into a path optimization challenge, providing a new perspective on the issue [5]. Yin et al. developed an enhanced disassembly hybrid graph (DHG) model tailored to EOL mobile phone components, specifically addressing the complexities of DSP [6]. Wu et al. streamlined the gravitational search algorithm, integrating a fast feasible solution generator for initial population creation, a precedence-preserving operator for subsequent generation development, and a multipoint optimization operator to explore neighboring solutions. An escape operator was also implemented to circumvent local optima traps, significantly bolstering the algorithm's problem-solving capabilities [7]. Zhang et al. refined the social engineering optimizer by incorporating an exchange operator, thereby enhancing the algorithm's solution efficiency [8].

Furthermore, Xing et al. used an enhanced ant colony algorithm to tackle the parallel DSP challenge, effectively demonstrating its efficacy through a case study involving a bevel gear reducer [9]. Xie et al. introduced an upgraded grey wolf optimizer (GWO), integrating three innovative operators to ensure the feasibility of its solutions under intricate constraints. They substantiated the effectiveness of these operators through two engineering case studies [10]. Liu et al. crafted a neighborhood search algorithm, incorporating four distinct neighborhood structures to refine the generation of feasible solutions [11]. Yu et al. employed a refined discrete whale optimization algorithm for DSP problems, augmenting it with a local search strategy to bolster the algorithm's performance [12]. Sun et al. developed a metaheuristic algorithm grounded in multi-neighborhood search strategies, tailored to address the parallel DSP problem and enhance solution quality [13]. Fu et al. proposed an improved GWO algorithm featuring a self-renewal mechanism and an exchange of optimization operators, both aimed at enhancing the algorithm's efficiency and stability [14].

The complexity of certain products and the interconnection of components can lead to sequence dependence during disassembly, where tasks without a predefined priority may interfere with one another. This interference can impede the most convenient disassembly of the prioritized task, potentially increasing disassembly time. Neglecting sequence dependence could lead to the need for rework, escalating the time and cost and impacting disassembly line efficiency. Consequently, the sequence-dependent DSP problem has emerged as a burgeoning field of scholarly interest.

In terms of the sequence-dependent DSP problem, Ren et al. leveraged an AOG to address the problem, aiming to maximize recovery profits by considering sequence dependence. Kim et al. tackled a sequence-dependent DSP problem by employing an extended flowchart to depict all potential disassembly sequences and crafting an integer programming model to address the problem effectively [15]. Ma et al. enhanced an AOG and proposed a two-stage algorithm to explore a sequence-dependent DSP problem, using an automatic pencil and a telephone as illustrative case studies [16]. Guo et al. introduced a multi-objective sequence-dependent DSP framework and employed a decentralized search approach to resolve their proposed challenges [17]. Xia et al. developed a model for partial DSP that accounts for sequence dependence in an uncertain environment, integrating particle swarm optimization with genetic algorithms to solve the problem [18]. Hartono et al. addressed a robot disassembly problem with sequence dependence by utilizing the BA to identify the optimal solution [19]. We have summarized the related literature from recent years, as shown in Table 1 [20–31].

Table 1. Related papers dedicated to DSP.

Recent Publications	Number of Objectives		Type of Decision Criteria		Consideration of Uncertainty		Consideration of Sequential Dependencies	
	Single	Multiple	Economic	Environmental	Yes	No	Yes	No
Yang et al. (2024) [20]		√	√	√		√		√
Hu et al. (2024) [21]		√	√			√		√
Chen et al. (2024) [22]		√	√	√		√		√
Liu et al. (2023) [23]		√	√	√	√			√
Zhang et al. (2024) [24]		√	√	√	√			√

Table 1. *Cont.*

Recent Publications	Number of Objectives		Type of Decision Criteria		Consideration of Uncertainty		Consideration of Sequential Dependencies	
	Single	Multiple	Economic	Environmental	Yes	No	Yes	No
Wang et al. (2023) [25]		√	√	√		√		√
Hartono et al. (2023) [26]		√	√	√		√		√
Liu et al. (2023) [27]	√		√		√			√
Gulivindala et al. (2023) [28]		√		√		√		√
Zhan et al. (2023) [29]		√		√		√		√
Liao et al. (2023) [30]		√	√		√			√
Qiu et al. (2022) [31]		√	√			√		√
This work		√	√	√	√		√	

Based on the above analysis of the literature, we have obtained the following insights:

(1) The current literature has not yet delved deeply into the uncertainties inherent in the sequence-dependent DSP problem. The subjects of disassembly, the products themselves, are fraught with unpredictability. Products may arrive in various compromised conditions, such as deformed, corroded, or contaminated, all of which can significantly influence the disassembly process. Furthermore, a disassembly operation is inherently variable. The duration required to complete a disassembly task is shaped by a variety of factors. These encompass the capabilities of the disassembly tools in use, the proficiency of the workers, and the conditions of their work environment, including their emotional state and levels of fatigue. To bolster the efficiency of disassembly operations, it is essential to conduct a thorough investigation into the sequence-dependent DSP problem within an uncertain context.

(2) Metaheuristic algorithms stand as the foundation for tackling DSP challenges. However, the well-established "no free lunch" theorem of optimization underscores the fact that no single algorithm is universally applicable. This reality necessitates an ongoing pursuit of innovation and refinement in algorithmic development. In alignment with this perspective, this study introduces an innovative bees algorithm (IBA) as a novel approach to address the DSP problem.

To address the challenges discussed, the key contributions of this paper are outlined below:

(1) This paper presents an analysis of the sequence-dependent DSP problem in an uncertain setting for the first time. We apply a stochastic simulation method to manage the uncertainties encountered during disassembly, more closely aligning our solutions to practical scenarios.

(2) We formulate a mathematical model that addresses the core challenges of the sequence-dependent DSP problem. This model aims to reduce both the time and energy required for disassembly.

(3) Considering the problem's complexity, we introduce an IBA in this study. Its performance is validated by comparing it with other advanced algorithms.

The structure of this paper is as follows: Section 2 outlines the modeling approach we have adopted and describes the development of a mathematical model for the sequence-dependent DSP problem that is designed to operate in uncertain environments. Section 3 offers a comprehensive explanation of our adapted IBA. Section 4 presents an examination of a real-world industrial scenario, along with a discussion and analysis of our findings. Section 5 illustrates the effectiveness of our proposed algorithm through comparative evaluation. Concluding the paper, Section 6 recaps the findings, recognizes the study's limitations, and proposes potential avenues for future inquiry.

2. Proposed Problem

In this section, the concept of the sequence-dependent DSP problem is introduced in Section 2.1. To represent the complex relationships between the objects to be disassembled, we present the DHG in Section 2.2. Finally, in Section 2.3, we introduce the multi-objective sequence-dependent DSP model.

2.1. Sequence-Dependent DSP Problem

In real life, the relationship between disassembly tasks is complex. Generally, DSP solely accounts for the priority constraints between tasks; however, given the intricate structure of EOL products and the interconnectivity of their components, tasks lacking a priority relationship may still interfere with each other. This interference can render the high-priority tasks less conveniently disassembled, leading to increased operational time. Such interference is referred to as 'sequence dependence' [32]. Usually, the spatial location of parts, the variation of disassembly tools, and the worker's constraints are the main reasons for sequence dependence [33]. For instance, during the disassembly of a computer monitor, if the audio control module placed next to the monitor's mainboard has already been disassembled, the time required to disassemble the mainboard is relatively short. However, if the audio control module has not been removed, spatial obstacles can lead to inconvenience for the worker during disassembly or may even necessitate a change of disassembly tools. Therefore, if there is an interference relationship between two disassembly tasks, we apply the DHG to build a sequence-dependent constraint matrix and define the interference time as a random variable. This approach helps us to ensure the efficiency of the disassembly process, reduce repetitive work, and optimize resource allocation.

2.2. Disassembly Hybrid Graph

We selected the DHG as an effective method for modeling the relationships between disassembly tasks. Figure 1 illustrates a DHG which includes sequence dependence. In this graph, the numbers within the circles represent the disassembly tasks. The solid arrows indicate the priority relationships between tasks, and the dotted arrows show the sequence dependence between them. For instance, between Task 1 and Task 12, the presence of a sequence dependence is denoted by dotted arrows. Owing to the symmetrical nature of sequence dependence, if Task 1 impedes the convenient disassembly of Task 12, then Task 12 reciprocally similarly hinders Task 1.

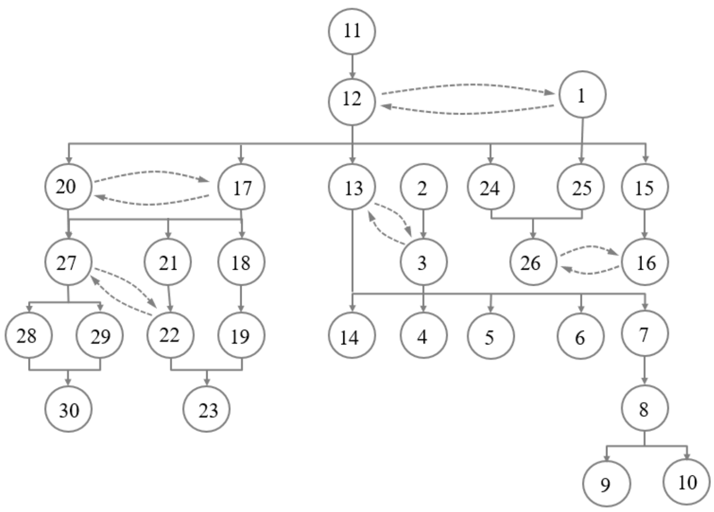

Figure 1. A DHG containing sequence dependence.

To undertake modeling calculations utilizing the DHG, it must be converted into its matrix representation. The given DHG can be represented by two matrices: the disassembly priority constraint matrix (P) and the sequence-dependent constraint matrix (Q). Taking Figure 2 as an example, if task 2 has a priority constraint relationship with task 3, this relationship is represented in the P matrix. This relationship is represented in the matrix as $P_3^2 = 1$. As shown in Figure 3, if there is an interference relationship between two tasks, such as task 1 and task 12, this relationship is represented in the Q matrix. This relationship is represented in the matrix as $Q_{12}^1 = 1$. Because of the symmetry of this relationship, this relationship is denoted as $Q_1^{12} = 1$.

$$
\mathrm{P} \left\{ \begin{matrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0
\end{matrix} \right.
$$

Figure 2. Disassembly priority matrix.

$$
\mathrm{Q} \left\{ \begin{matrix}
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{matrix} \right.
$$

Figure 3. Sequence-dependent matrix.

As seen in Figure 1, we can construct the disassembly priority constraint matrix P.

We can construct the sequence-dependent constraint matrix Q, as shown in Figure 3.

Utilizing the disassembly priority constraint matrix P, we ascertain the prerequisites for each disassembly task: a part must not have a priority part that has not yet been disassembled. Furthermore, the matrix Q is employed to address scenarios where there is interference despite the absence of a priority relationship between tasks. In such cases, it is necessary to consider the additional interference time that arises due to sequence dependence.

2.3. Proposed Model

As previously stated, to ensure that the disassembly solutions we obtain are more reflective of real-world conditions, it is essential to account for the inherent uncertainties involved. To achieve this, we have adopted an expectation model, an important application of probability theory, and a significant branch of stochastic programming. This approach is notably efficient and straightforward compared to other methodologies, offering a streamlined framework to predict and evaluate expected disassembly outcomes amidst uncertainty.

Furthermore, we aim to simultaneously optimize two critical objectives: disassembly time and energy consumption. Each of these objectives holds specific significance:

- Disassembly time: The swift disassembly of products helps to lessen the adverse environmental effects of waste, reducing the risks of hazardous substance spills and disruptions to ecosystems. It also enables the quicker recycling of valuable materials and components.
- Disassembly energy consumption: Minimizing energy use is a vital step towards a circular economy. This strategy not only reduces environmental harm but also allows businesses to cut operational costs, thus enhancing economic efficiency.

In terms of the notations used in our model, their meaning is as follows.

Indices:	
m, l	Index of disassembly tasks, $m,\ l \in \{1,\ 2, ...,\ M\}$
Parameters:	
M	Total number of tasks in the EOL product
t_m	Stochastic disassembly time for task m
t_t	Stochastic time to change the tool of disassembly
t_d	Stochastic time to change the direction of disassembly
e_m	Unit time of energy consumption of task m
e_t	Stochastic time to change the tool of disassembly
e_d	Stochastic time to change the direction of disassembly
g_m	Difficulty of disassembly task m
sd_{ml}	Stochastic increase in time when task m is interfered with by task l
p_{ml}	1, if task m must be executed before task l, otherwise 0
Decision variables:	
a_{ml}	1, if task m is executed before task l, otherwise 0
x_{ml}	1, if task m is interfered with by task l, otherwise 0
y_m	1, if the task m requires a different tool to the previous task in the sequence, otherwise 0
z_m	1, if the task m is in a different direction to the previous task in the sequence, otherwise 0

Then, our model is composed as follows:

$$f_1 = minE\left(\sum_{m=1}^{M}(t_m + t_t y_m + t_d z_m) + \sum_{m=1}^{M}\sum_{l=1}^{L} sd_{ml}x_{ml}\right) \tag{1}$$

$$f_2 = minE\left(\sum_{m=1}^{M}(1+g_m)e_m t_m + \sum_{m=1}^{M}(e_t y_m + e_d z_m)\right) \tag{2}$$

such that

$$\sum_{l=1}^{L} x_{ml} \leq 1,\ \ \forall m \in \{1,2,\ldots,M\} \tag{3}$$

$$x_{ml} + x_{ml} \leq 1,\ \ \forall m \in \{1,2,\ldots,M\},\ \forall l \in \{1,2,\ldots,M\} \tag{4}$$

$$a_{ml} \geq p_{ml},\ \ m = 1,2,\ldots,M,\ \ l = 1,2,\ldots,M \tag{5}$$

$$x_{ml}, y_m, z_m = \{0,1\},\ \ m = 1,2,\ldots,M,\ \ l = 1,2,\ldots,M \tag{6}$$

Equation (1) denotes the minimized disassembly time. In this paper, the total disassembly time is expressed as the sum of the product disassembly time, the direction change time, the tool change time, and the interference time due to sequence dependence. Equation (2) denotes the minimized disassembly energy consumption. The disassembly's energy consumption consists of three parts: the energy consumption generated by the disassembly task, the energy consumption generated by switching the direction of disassembly, and the energy consumption generated by changing the disassembly tool. Constraint (3) ensures that each disassembly task is executed only once. Constraint (4) ensures that the sequence dependence between tasks does not form a loop, so if task m is executed after task l, task l cannot be executed after task m. Constraint (5) states that for task m and l, if task m must be executed before task l, then a_{ml} must also be 1, ensuring that task m is in fact executed before task l. Constraint (6) defines the binary variables in the model. These constraints are essential to ensure that the disassembly process unfolds in a logical sequence.

3. Proposed Solution Method

To address the problem efficiently, we have adopted the IBA as our solution tool. The traditional BA simulates the intelligent foraging behavior of bees in nature. This approach is intuitive and adept at swiftly converging upon the global optimum, as reported in the literature [34,35]. The conventional BA is categorized into three main parts: the nectar source, scout bees, and foraging bees. Scout bees are categorized according to their fitness and are divided into optimal scout bees and better scout bees and the rest of the scout bees. Optimal scout bees and better scout bees send a larger number of follower bees to go out to forage for nectar together, while the rest of the scout bees search for new nectar sources through random global searches to ensure the diversity of nectar sources. Foraging bees do not actively go out to search for nectar sources, but wait for the scout bees to come back from foraging, and, after obtaining information about the nectar sources from the scout bees, they follow the scout bees to search for new nectar sources in the vicinity of the nectar sources. The best nectar source in this search is retained through a global search and neighborhood search and proceeds to the next iteration. The conventional BA is prone to fall into local optimal solutions in some cases, leading to local optimality in the search process. Therefore, we adopt a roulette strategy to dynamically adjust the neighborhood structure by adaptively updating the probability of scout bees being selected and operator weights to avoid falling into a local optimum during the search process.

In this section, we outline the core steps of the IBA. We start by detailing the population initialization strategy of the IBA, ensuring a diverse and robust initial population that provides a strong foundation for the algorithm's search process (Section 3.1). We then discuss the role assignment method used for the bees, a critical component of the algorithm (Section 3.2). Subsequently, we present novel search operators that are designed to refine existing solutions and enhance the optimization process, which is explained in Sections 3.3–3.5. Additionally, we introduce a constraint correction strategy to ensure all solutions generated adhere to the established constraints (Section 3.6). We also proceed to discuss the population update mechanism and the termination conditions of the algorithm, which are crucial for ensuring both efficiency and accuracy (Section 3.7). After a thorough analysis of the components of the IBA, we provide pseudo-code and flowchart of the algorithm to provide a clear view of the internal logic of the IBA.

3.1. Population Initialisation

In metaheuristic algorithms, population initialization is a key step to ensuring the effectiveness of the algorithm [36]. Before delving into the generation of feasible initial solutions, we initially present the encoding method employed in this paper. We use real number encoding to represent the disassembly sequence; assuming that an EOL product consists of M disassembly tasks, the disassembly initial sequence can be represented as $S = (a_m), i = 1, 2, \cdots M$.

Then, to ensure that the execution order of the disassembly tasks complies with the predefined constraints, thereby generating a feasible initial solution, we utilize the priority matrix P in the generation of the disassembly sequence. We commence by identifying tasks within the matrix that are not restricted by priority constraints. Subsequently, we randomly select one of these tasks and incorporate it into our initial solution, and then we update the priority matrix P to reflect this selection. This iterative process of identifying, selecting, and updating is continuously repeated until we successfully construct a feasible initial solution that aligns with the predefined constraints, thereby setting a solid foundation for the optimization process that follows.

3.2. Classification of Scout Bees' Role

In the formulation of the IBA, the assignment of scout bee roles is a pivotal step, similar to its counterpart in the BA. We initiate the process by creating a population of g_{size} scout bees. This population is then subjected to an evaluation through fast non-dominated sorting, complemented by the calculation of crowding distance [37].

Fast non-dominated sorting is a technique employed in multi-objective optimization problems which is designed to categorize individuals in a population into hierarchical levels based on non-domination relationships. In such problems, a solution is considered non-dominated by another if it is not inferior across all objectives and is superior in at least one objective. This sorting method allows us to segregate the feasible solutions into distinct Pareto frontiers.

The crowding distance is applied within each Pareto frontier, calculating the proximity of each solution to its nearest neighbors. This metric reflects the concentration of individuals within a rank layer post non-dominated ranking, offering insight into the diversity of the possible solutions.

By combining fast non-dominated sorting with the calculation of crowding distance, we can effectively rank all generated initial solutions. This ranking allows us to efficiently identify the optimal and sub-optimal scout bees based on their order. Meanwhile, the remaining scout bees are categorized as random scout bees. It should be noted that, following the ranking process, we designate the top NS scout bees from the sorted list as the optimal scout bees. We then identify the next ES scout bees as the better scout bees. The remaining random scout bees are defined as RS.

It should be noted that since the fast non-dominated sorting and crowding distance calculation are based on the objective function values (f_1, f_2) of each solution, our model includes random variables. To obtain objective function values for each disassembly sequence scheme, we use the Monte Carlo simulation method. Our specific approach is that, for each scheme, we perform 500 Monte Carlo simulations. In each simulation process, the random variables are sampled randomly according to their probability distribution functions, thus converting them into precise values. Ultimately, we average the objective function values obtained from the 500 simulations for each scheme to determine its final set of objective function values.

3.3. Search Phase of the Optimal Scout Bees

After the scout bees are assigned roles, the IBA proceeds to the phase of the optimal scout bees' search, where the foraging bees follow the optimal scout bees to search for nectar around them, which can be seen as a neighborhood search around the optimal scout bees. The number of foraging bees they carry is equivalent to the number of neighborhood searches conducted. In this paper, the number of foraging bees carried by the optimal scout bees is set to OF. In this phase, we introduce two main search operators, as shown in Figures 4 and 5. In the figure, blue denotes the points selected by the operator, yellow represents the unselected points, and the blue arrows signify the movement direction of the selected points. We choose between these two operators through a roulette strategy and apply it to the neighborhood search of the optimal and better bees to find a better feasible solution.

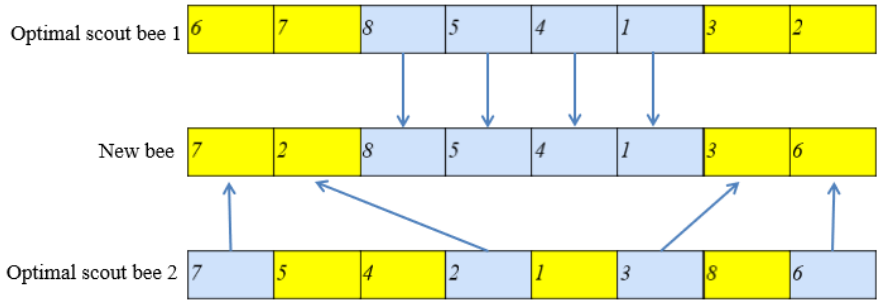

Figure 4. OX operator.

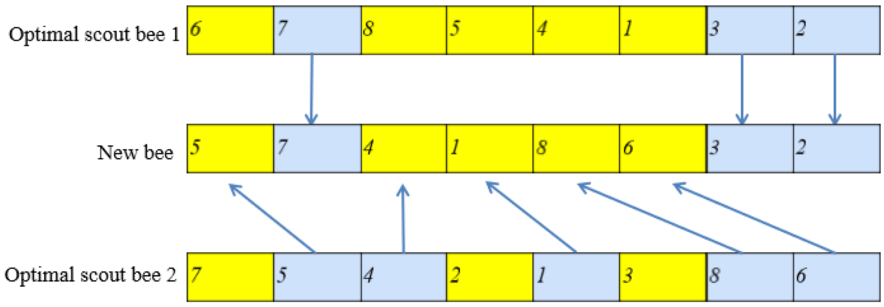

Figure 5. PBX operator.

(1) Order crossover (OX)

We randomly select another optimal scout bee in addition to the current optimal one. A few consecutive points are selected from optimal scout bee 1, these selected points are placed in the same position in the new bee, and, finally, the remaining points in optimal scout bee 2 are inserted sequentially into the new bee.

(2) Position-based crossover (PBX)

We randomly select another optimal scout bee in addition to the current optimal one. A few points are randomly selected from optimal scout bee 1, these selected points are placed in the same position in the new bee, and, finally, the remaining points in optimal scout bee 2 are inserted into the new bee in order. Unlike the OX operator, the position of the selected points in this method can be unfixed, as shown in Figure 5 below.

We construct the roulette wheel calculation rule as follows:

Step 1: The initial weights of both operators are set to 1 and their scores are set to 0 for the first iteration, and thus the probabilities of the operators being selected are as follows:

$$P_{S(i)} = \frac{W_{S(i)}}{\sum_j W_{S(i)}}, \ \ j \in (1, 2, \cdots \mathrm{J}) \tag{7}$$

where $W_{S(i)}$ denotes the operator weights, $P_{S(i)}$ denotes the probability that the operator is selected, and J denotes the maximum number of iterations.

Step 2: In the iterative process of the algorithm, different scores are given to the operator according to the quality of the updated solution after each iteration, and scores are set in decreasing order. The scoring rule is as follows:

$$S = \left\{ \begin{matrix} S1 \ \text{ if the new solution is replaced by best} - \text{known solution} \\ S_2 \text{ other} \end{matrix} \right\} \tag{8}$$

Step 3: The weights of the operators are set to be updated every m generations according to the operator scores, and the formula for this is as follows.

$$W_{S(i)}{}^{j+1} = \left\{ \begin{array}{l} W_{S(i)}{}^{j}, \mu_0 = 0 \\ (1-\rho) W_{S(i)}{}^{j} + \rho \frac{S}{\mu_0}, \mu_0 > 0 \end{array} \right. \tag{9}$$

where $W_{S(i)}{}^{j}$ is the operator weight of jth, μ_0 is the number of times the operator has been selected, S is the cumulative score of the operator in this iteration, and ρ is the weight adjustment factor. Through this formula, the operator weights are linked to their historical performance to achieve the aim of adaptive operator selection.

3.4. Search Phase of the Better Scout Bees

Subsequently, the IBA proceeds with the exploration phase of the better scout bees. Its fundamental principle mirrors that of the optimal scout bees' search phase. We designate the number of foraging bees that follow the better scout bees as NF. In this phase, we design two search operators, as shown in Figures 6 and 7. The blue colour in the figure represents the points selected by the operator, the green colour denotes the unselected points, and the arrows signify the direction of movement for the points.

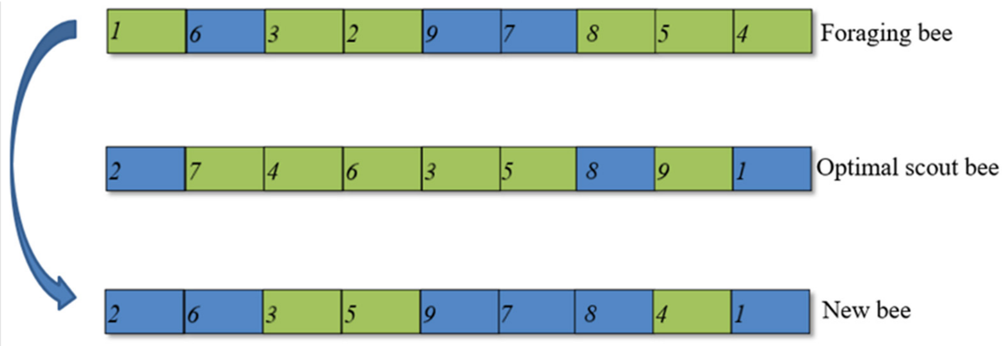

Figure 6. S1 operator.

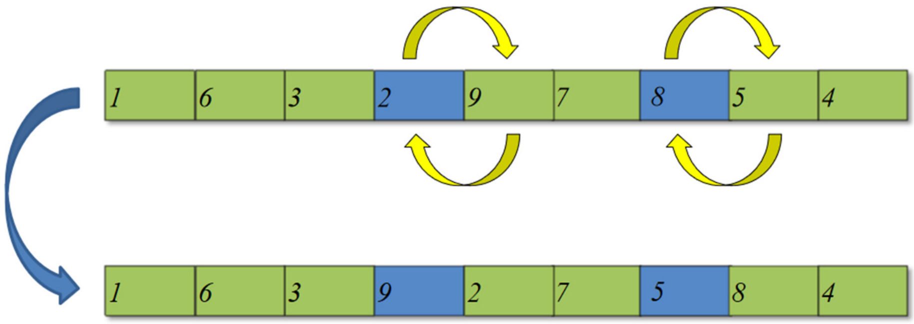

Figure 7. S2 operator.

(1) Randomly select n points in the optimal scout bee and a different n points in the foraging bee. Insert these points into the new bee in the same order, and insert the remaining points of the foraging bee into the new bee in the same order, as shown in Figure 6.

(2) Randomly select two points in the better scout bee and cross them with the next one, or with the previous one if the selected point is the last one, as shown in Figure 7 below.

We construct the roulette wheel calculation rule as follows:

Step 1: Calculate the probability of the optimal scout bee being selected, as demonstrated in Equation (10).

$$P_{(N(i))} = \frac{SF_{(Nsize(i))}}{\sum_{i=1}^{Nsize} SF_{(N(i))}}, i = 1,\ 2,\ 3,\ Nsize \tag{10}$$

where $P_{(N(i))}$ is the probability of the optimal scout bee being selected as the ith bee. $SF_{(N(i))}$ is the fitness value of the ith beet, and $Nsize$ is the total number of bees.

Step 2: Calculate the cumulative probability of each optimal scout bee, as demonstrated in Equation (11).

$$q(i) = \sum_{j=1}^{i} P_{(N(j))},\ i = 1,\ 2,\ 3, N \quad (11)$$

where $q(i)$ is the cumulative probability of the ith optimal scout bee.

Step 3: Select the optimal scout bee.

We set the number of foraging bees following the better scout bee to NF. Then, each better scout bee will generate NF new solutions according to the above operator. We compare these NF new solutions, select the best solution after calcultating the crowding distance calculation, and save it.

3.5. Constraint Correction Methods

In the domain of metaheuristic algorithms, newly formulated solutions can occasionally fail to conform to predefined constraints. This challenge is also encountered within the proposed IBA. For the newly generated disassembly sequence, we start checking for this from the first task. If it satisfies the constraints of executing the disassembly precedence relation and the sequential dependency relation, we continue to check the subsequent tasks. If one of them is not executable, a randomly generated executable task is used instead, and then the executability of the next task is checked until the last task satisfies the disassembly requirements.

3.6. Termination of the Algorithm

When all scout bees have completed their search, the newly generated solution is merged with the original scout bee population. Then, fast non-dominated sorting and crowding distance calculations are performed to update the scout bee population and assign new scout bee roles. In addition, the set of non-dominated solutions is stored in an external archive which is updated after each iteration. After reaching the maximum number of iterations *Maxit* or a set stopping time, the non-dominated solution is exported and placed in the external archive.

3.7. Algorithmic Framework

In this section, we provide the flowchart (Figure 8) and pseudo-code (Algorithm 1) of the proposed IBA, which clearly show the detailed process of the algorithm population from its initialization to the end of its iterations.

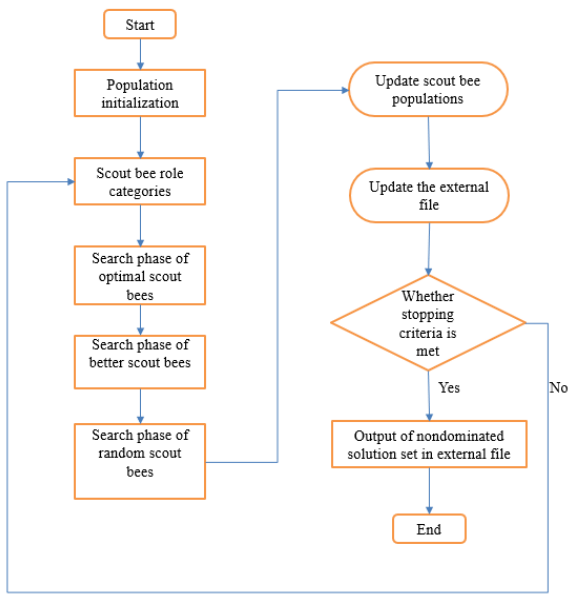

Figure 8. IBA flowchart.

Algorithm 1: IBA Main Loop

```
Input: Algorithm parameters, problem parameters
Output: Pareto non-dominated solution set
For i = 1:g_size
Produce scout bee individuals, as shown in Section 3.1
End For
Form optimal scout bees, better scout bees, and random scout bees, as shown in Section 3.2
it = 0
While it < Maxit
For i = 1:NS #Search phase of optimal scout bees #
For j = 1:OF
Select operator, as shown in Section 3.3
Search for nectar near present scout bee, as shown in Section 3.3
Update operator weights
End For
Save non-dominated nectar
Reset operator weights
End For
For i = 1:ES # Search phase of better scout bees #
For j = 1:NF
Select operator, as shown in Section 3.3
Search for nectar near present scout bee, as shown in Section 3.4
Update operator weights
End For
Save non-dominated nectar
Reset operator weights
End For
For i = 1:RS # Search phase of random scout bees #
Randomly generate new scout bees, as shown in Section 3.1
End For
Update scout bees population
Save non-dominated solutions to an external archive
it = it + 1
End While
Obtain the non-dominated solution set of all solutions saved in the external archive
Output the final non-dominated solution set
```

4. Case Study

In this section, to validate the effectiveness of the algorithm proposed in this paper, we used the real industrial case presented by Shan [38]. Table 2 enumerates the relevant data pertaining to the disassembly process of a lithium battery. In the table, the geometric centre of the lithium battery is selected as the coordinate origin, the three-dimensional coordinate method is used to label the disassembly direction, and U denotes the random disassembly time in the interval. All the data and code used for the metaheuristic algorithm were executed on the MATLAB software R2013a, and the algorithm was run on an Intel(R) Core(TM) i5-9300H CPU at 2.40 GHz, with 8 GB of RAM.

Based on the product's disassembly relationship and spatial location constraints, the DHG of the lithium battery can be obtained, as shown in Figure 9. The solid arrows in the figure indicate disassembly priority relationships between tasks, and the dotted arrows indicate sequence dependence between tasks.

Table 2. Lithium battery's specific disassembly information.

Order	Name	Tool	Direction	Disassembly Time
1	Fastening screws around the cover	wrench	+Z	U (175, 182)
2	Fastening screws in the center of the cover	wrench	+Z	U (54, 56)
3	Repair switch	wrench	+Z	U (43, 44)
4	Maintenance switch fastening screws	Screwdriver	+Z	U (28, 32)
5	Connecting plate fastening screws	Screwdriver	+Z	U (42, 47)
6	Box cover	wrench	+Z	U (20, 25)
7	Copper Cable Ties	wrench	+Z	U (58, 61)
8	Pipe Ties	Plier	+Y	U (42, 47)
9	Wire Harness Tie	Plier	+Y	U (40, 43)
10	Copper Tape	Plier	+Y	U (16, 19)
11	Wiring Harness	Plier	+Y	U (8, 10)
12	Wire Harness Plugs	Plier	+Y	U (32, 35)
13	Copper Protection Shell	Plier	+Y	U (18, 22)
14	Copper fastening screws	Plier	+Y	U (21, 25)
15	Copper busbar	Hand	-Y	U (10, 13)
16	Battery Management System	Hand	-Y	U (22, 24)
17	Battery Management System fastening screws	Plier	-Y	U (21, 24)
18	Charging equipment cover	Plier	-Z	U (14, 16)
19	Charging equipment bottom	Plier	-Z	U (4, 6)
20	Screws for the bottom of the charging unit	wrench	+Z	U (16, 50)
21	Charging equipment base plate	wrench	-Y	U (34, 37)
22	Charging equipment base plate fastening screws	wrench	-Y	U (23, 25)
23	Shims	Plier	-Y	U (15, 19)
24	Gasket fastening screws	Plier	-Y	U (45, 50)
25	Current Sensing Wire fastening screws	Screwdriver	-X	U (27, 30)
26	Relay Plugs	Screwdriver	-X	U (15, 20)
27	Current Sensor	Screwdriver	-X	U (30, 35)
28	Relay	Screwdriver	+X	U (40, 45)
29	Fuses	Screwdriver	+X	U (25, 30)
30	Current Sensor Fastening Screws	Hand	+X	U (8, 12)
31	Relay Fastening Screws	Hand	+Y	U (17, 20)
32	Fuse fastening screws	Hand	+Y	U (8, 15)
33	Adapter plate	Screwdriver	+Z	U (8, 10)
34	Splice plate fastening screws	wrench	+Z	U (18, 25)
35	Module fastening screws	Plier	+Z	U (17, 19)
36	Module Fastener	Plier	+Z	U (3, 7)
37	Module 1	Plier	-Z	U (2, 5)
38	Module 2	wrench	-Z	U (14, 17)
39	Coolant Tube Snap	wrench	-Y	U (15, 20)
40	Coolant Plastic Tube	wrench	-Y	U (4, 6)
41	Condensate tube fastening screws	wrench	-Y	U (12, 25)
42	Condensate tube	Screwdriver	-Y	U (14, 17)
43	Thermal Conductive Silicone	Screwdriver	-Y	U (20, 25)
44	Bottom	Screwdriver	-Y	U (4, 8)

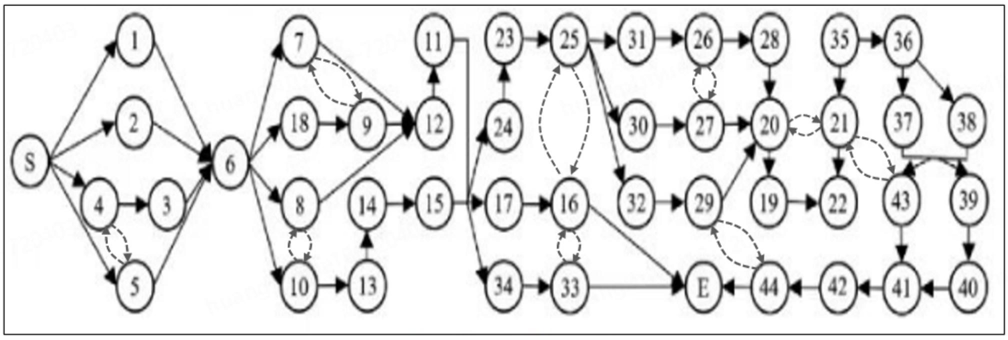

Figure 9. DHG of lithium battery disassembly, containing 44 tasks.

Based on the pre-experiment, and considering the balance of solving time and quality, the parameters for the IBA are defined in Table 3.

Table 3. IBA parameter setting.

Parameters	Value
$Maxit$	200
g_{size}	50
NS	8
ES	5
OF	6
NF	5

Upon completion of a single iteration, the IBA successfully produced eight sets of non-dominated solutions, which are detailed in Table 4.

Table 4. Non-dominated solution sets.

Order	Non-Dominated Solutions	f_1	f_2
1	[5,35,36,38,2,37,39,43,1,40,21,41,42,4,3,6,10,44,8,18,9,7,12,13,14,11,15,17,24,23,16,34,33,25,31,32,29,30,26,28,27,20,19,22]	1575.78	1893.11
2	[5,35,2,36,38,37,39,43,1,40,21,41,42,4,44,3,6,8,10,18,9,7,12,13,14,11,15,17,24,23,16,34,33,25,31,32,29,30,26,28,27,20,19,22]	1548.95	1934.45
3	[5,35,2,36,38,37,21,43,1,4,3,6,18,9,10,8,39,7,12,13,11,14,40,15,34,17,41,16,24,23,25,30,33,32,31,42,29,44,26,28,27,20,19,22]	1639.06	1684.36
4	[5,35,2,36,38,37,39,43,1,40,21,41,42,4,3,6,10,44,8,18,9,7,12,13,14,11,15,17,16,24,23,34,33,25,31,32,29,30,26,28,27,20,19,22]	1530.31	1947.11
5	[5,35,2,4,3,36,38,1,6,10,21,7,37,43,39,40,13,41,8,18,9,14,12,11,15,34,33,17,24,23,16,42,25,31,44,32,29,30,26,28,27,20,19,22]	1621.75	1778.63
6	[5,35,2,36,38,37,39,43,1,40,21,41,42,4,3,6,8,44,10,18,9,7,12,13,14,11,15,17,24,23,16,34,33,25,31,32,29,30,26,28,27,20,19,22]	1573.78	1906.38
7	[5,35,2,36,38,37,39,43,1,40,21,41,42,4,3,6,10,44,8,18,9,7,12,13,14,11,15,17,24,23,34,16,33,25,31,32,29,30,26,28,27,20,19,22]	1548.99	1923.55
8	[5,35,2,36,38,37,39,1,43,40,21,41,42,4,3,6,10,44,8,18,9,7,12,13,14,11,15,17,24,23,34,16,33,25,31,32,29,30,26,28,27,20,19,22]	1512.80	1991.95
9	[5,35,2,36,38,1,37,43,4,21,39,40,3,6,10,18,7,41,8,42,9,12,11,13,14,15,44,17,24,23,16,34,33,25,31,32,29,30,27,26,28,20,19,22]	1528.48	1970.61
10	[5,35,2,36,38,4,37,43,1,3,21,6,10,13,18,8,7,14,9,39,40,12,15,41,11,42,24,17,16,23,25,34,33,31,30,32,29,27,26,28,44,20,19,22]	1720.40	1574.99

An analysis of Table 4 reveals a trade-off between the objectives of minimizing disassembly time and energy consumption. For decision-makers prioritizing the reduction in disassembly time, Solution 8 emerges as the optimal choice. However, this solution is associated with the highest level of energy consumption. Conversely, for those primarily concerned with minimizing energy usage, Solution 10 is the most effective, albeit at the expense of a longer disassembly time. Consequently, decision-makers must weigh the relative importance of these two factors to select a disassembly solution that aligns with their specific requirements.

5. Comparison with Other Algorithms

Our proposed IBA is easy to implement, can be applied to multi-mode searches, and is not prone to falling into local optimal solutions. However, other heuristic algorithms have a high solution efficiency in solving multi-objective optimization problems, such as NSGA-II, which can gradually improve its results and increase its convergence rate through multiple iterations. Therefore, we evaluated the performance of the IBA by benchmarking it against established algorithms recognized for their efficacy in the literature. These include the non-dominated sorting genetic algorithm II (NSGA-II) [39], the improved immune algorithm (IA) [40], and the innovative multi-objective enhanced water wave optimization (EWWO) algorithm [41]. To ensure a thorough performance assessment, we employed three multi-objective evaluation metrics: the number of Pareto solutions (NPSs), inverted generational distance (IGD), and hypervolume (HV). The NPSs metric quantifies the diversity of the Pareto solutions, with a higher value indicating a more effective metaheuristic in generating a comprehensive set of solutions. The IGD metric measures the proximity of the non-dominated solution set to the true Pareto frontier, with a lower value suggesting better convergence and distribution [42]. The HV metric, introduced by Zitzler et al., assesses the volume covered by the non-dominated solutions, providing insight into the solution set's quality and diversity [43]. These metrics enable a comprehensive comparison of the IBA's performance with other advanced algorithms

To ensure a fair comparison, all algorithms were given a uniform running time of 120 s and a population size of 50, other parameters were configured using the relevant literature. Consistent coding methodologies were employed to mitigate the effects of heuristic randomness. Each algorithm was executed 15 times, and their average results are presented in Table 5. The data indicate that the IBA achieves optimal values across the NPSs, HV, and IGD metrics, signifying its ability to identify a broader range of Pareto solutions that are well-distributed across the objective space. Collectively, these results underscore the IBA's robust performance in multi-objective optimization, delivering solution sets characterized by their quality, diversity, compactness, and stability.

Table 5. Performance of the algorithms.

Algorithm	NPS	HV	IGD
IBA	12.62	0.74	0.12
NSGA-II	10.54	0.65	0.14
IA	9.87	0.73	0.12
EWWO	8.78	0.66	0.13

The boxplot in Figure 10 illustrates the above results.

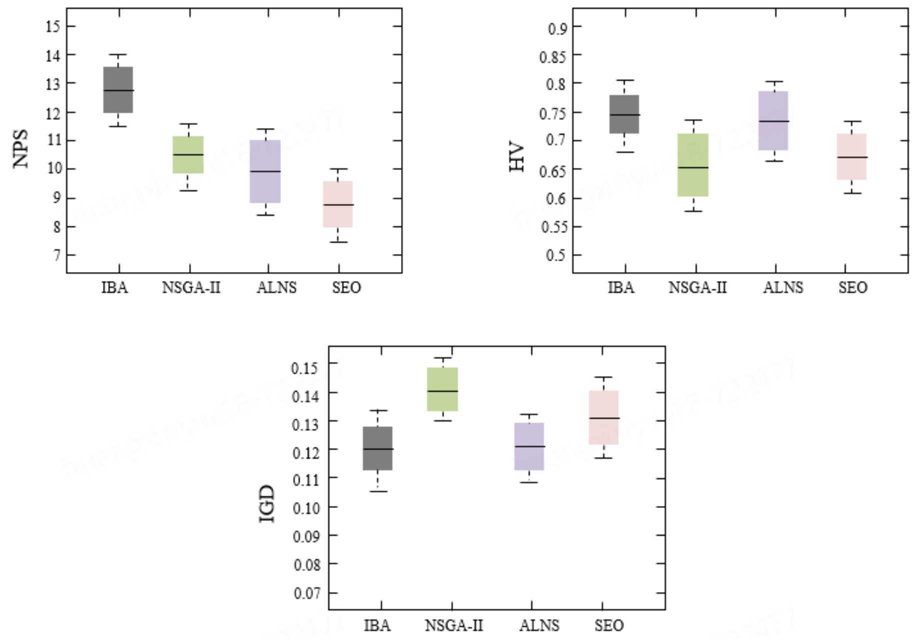

Figure 10. Statistical results of algorithm comparison in terms of NPSs, HV, and IGD.

6. Conclusions and Future Work

In this study, we explore the sequence-dependent DSP problem within uncertain environments. Recognizing the influence of uncertainty on disassembly efficiency, we strategically employ an expectation model to effectively manage these uncertainties. Furthermore, we develop a multi-objective assessment model that incorporates factors like disassembly time and energy consumption for a holistic evaluation of the proposed problem. To address the problem efficiently, we introduce the IBA, which features an effective population initialization strategy and innovative search operators. A case study further confirms the trade-offs among its objectives and offers decision-makers solutions arrived at from various perspectives. Furthermore, the IBA is compared with other state-of-the-art algorithms, further proving its effectiveness and superiority, as it has certain advantages in the NPSs, IGD, and HV metrics.

The results show that this method can find non-dominated disassembly sets with a good convergence and diversification performance. Given the current social situation of energy reduction, effective disassembly management can reduce the impact of waste on the environment and promote the recycling of resources. By optimizing the dismantling process, managers can be prompted to carry out long-term strategic planning, consider the environmental impact of the whole life cycle of products, and promote the transition of enterprises to a circular economy [44]. This study advances the resolution of the sequence-dependent DSP problem under uncertainty, but we recognize its limitations. Other uncertainties in the actual disassembly process, such as equipment breakdowns and operator skill variations, require more in-depth investigation in future studies. Moreover, although the IBA has been customized to solve specific issues within this study's scope, the algorithm's applicability to a wider array of optimization challenges requires ongoing exploration and empirical validation.

Author Contributions: Data curation, X.Z.; Formal analysis, C.Z.; Investigation, X.Z.; Methodology, X.H.; Validation, C.Z. and Y.G.; Writing—original draft, X.H.; Writing—review and editing, X.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data are contained within the article.

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

References

1. Zhang, X.; Zhou, H.; Fu, C.; Mi, M.; Zhan, C.; Pham, D.T.; Fathollahi-Fard, A.M. Application and planning of an energy-oriented stochastic disassembly line balancing problem. *Environ. Sci. Pollut. Res.* **2023**, 1–15. [CrossRef]
2. Tang, Y.; Zhou, M.; Zussman, E.; Caudill, R. Disassembly modeling, planning and application: A review. In Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), San Francisco, CA, USA, 24–28 April 2000; IEEE: Piscataway, NJ, USA, 2002; Volume 3, pp. 2197–2202.
3. Ren, Y.; Meng, L.; Zhang, C.; Zhao, F.; Saif, U.; Huang, A.; Mendis, G.P.; Sutherland, J.W. An efficient metaheuristic for sequence-dependent disassembly planning. *J. Clean. Prod.* **2020**, *245*, 118644. [CrossRef]
4. Li, J.R.; Khoo, L.P.; Tor, S.B. A novel representation scheme for disassembly sequence planning. *Int. J. Adv. Manuf. Technol.* **2002**, *20*, 621–630. [CrossRef]
5. Wang, H.; Xiang, D.; Duan, G. A genetic algorithm for product disassembly sequence planning. *Neurocomputing* **2008**, *71*, 2720–2726.
6. Yin, F.; Wang, K.; Wang, X.; Li, L.; Liu, G.; Maani, T.; Sutherland, J.W. An improved disassembly hybrid graph model for selective disassembly sequence planning. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2024**, *238*, 1519–1530. [CrossRef]
7. Wu, P.; Wang, H.; Li, B.; Fu, W.; Ren, J.; He, Q. Disassembly sequence planning and application using simplified discrete gravitational search algorithm for equipment maintenance in hydropower station. *Expert Syst. Appl.* **2022**, *208*, 118046. [CrossRef]
8. 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]
9. Xing, Y.; Wu, D.; Qu, L. Parallel disassembly sequence planning using improved ant colony algorithm. *Int. J. Adv. Manuf. Technol.* **2021**, *113*, 2327–2342. [CrossRef]
10. Xie, J.; Li, X.; Gao, L. Disassembly sequence planning based on a modified grey wolf optimizer. *Int. J. Adv. Manuf. Technol.* **2021**, *116*, 3731–3750. [CrossRef]
11. Liu, H.; Zhang, L. Optimizing a disassembly sequence planning with success rates of disassembly operations via a variable neighborhood search algorithm. *IEEE Access* **2021**, *9*, 157540–157549. [CrossRef]
12. Yu, D.; Zhang, X.; Tian, G.; Jiang, Z.; Liu, Z.; Qiang, T.; Zhan, C. Disassembly Sequence Planning for Green Remanufacturing Using an Improved Whale Optimisation Algorithm. *Processes* **2022**, *10*, 1998. [CrossRef]
13. Sun, X.; Guo, S.; Guo, J.; Du, B.; Tang, H. An improved multi-objective evolutionary algorithm for multiple-target asynchronous parallel selective disassembly sequence planning. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2023**, *237*, 1553–1569. [CrossRef]
14. Fu, W.; Liu, X.; Chu, F.; Li, B.; Gu, J. A disassembly sequence planning method with improved discrete grey wolf optimizer for equipment maintenance in hydropower station. *J. Supercomput.* **2023**, *79*, 4351–4382. [CrossRef]
15. Kim, H.W.; Lee, D.H. An optimal algorithm for selective disassembly sequencing with sequence-dependent set-ups in parallel disassembly environment. *Int. J. Prod. Res.* **2017**, *55*, 7317–7333. [CrossRef]
16. Ma, Y.S.; Jun, H.B.; Kim, H.W.; Lee, D.H. Disassembly process planning algorithms for end-of-life product recovery and environmentally conscious disposal. *Int. J. Prod. Res.* **2011**, *49*, 7007–7027. [CrossRef]
17. Guo, X.; Zhou, M.; Liu, S.; Qi, L. Lexicographic multiobjective scatter search for the optimization of sequence-dependent selective disassembly subject to multiresource constraints. *IEEE Trans. Cybern.* **2019**, *50*, 3307–3317. [CrossRef] [PubMed]
18. Xia, X.; Liu, W.; Zhang, Z.; Wang, L. Partial disassembly line balancing problem analysis based on sequence-dependent stochastic mixed-flow. *J. Comput. Inf. Sci. Eng.* **2020**, *20*, 061005. [CrossRef]
19. Hartono, N.; Ramírez, F.J.; Pham, D.T. Optimisation of Product Recovery Options in End-of-Life Product Disassembly by Robots. *Automation* **2023**, *4*, 359–377. [CrossRef]
20. Yang, S.; Zhuo, X.; Ning, W.; Xia, X.; Huang, Y. Integrated Risk-Aware Smart Dis-assembly Planning for Scrap Electric Vehicle Batteries. *Energies* **2024**, *17*, 2946. [CrossRef]
21. Hu, Y.; Liu, C.; Zhang, M.; Lu, Y.; Jia, Y.; Xu, Y. An ontology and rule-based method for human–robot collaborative disassembly planning in smart remanufacturing. *Robot. Comput.-Integr. Manuf.* **2024**, *89*, 102766. [CrossRef]
22. Chen, Z.; Cheng, H.; Liu, Y.; Aljuaid, M. An improved artificial bee colony algorithm for the multi-objective cooperative disassembly sequence optimization problem considering carbon emissions and profit. *Eng. Optim.* **2024**, 1–22. [CrossRef]
23. Liu, J.; Xu, Z.; Xiong, H.; Lin, Q.; Xu, W.; Zhou, Z. Digital twin-driven robotic dis-assembly sequence dynamic planning under uncertain missing condition. *IEEE Trans. Ind. Inform.* **2023**, *19*, 11846–11855. [CrossRef]
24. Zhang, X.; Fu, A.; Zhan, C.; Pham, D.T.; Zhao, Q.; Qiang, T.; Aljuaid, M.; Fu, C. Selective disassembly sequence planning under uncertainty using trapezoidal fuzzy numbers: A novel hybrid metaheuristic algorithm. *Eng. Appl. Artif. Intell.* **2024**, *128*, 107459. [CrossRef]
25. Wang, K.; Guo, J.; Du, B.; Li, Y.; Tang, H.; Li, X.; Gao, L. A novel MILP model and an improved genetic algorithm for disassembly line balancing and sequence planning with partial destructive mode. *Comput. Ind. Eng.* **2023**, *186*, 109704. [CrossRef]
26. Hartono, N.; Ramírez, F.J.; Pham, D.T. A multiobjective decision-making approach for modelling and planning economically and environmentally sustainable robotic disassembly for remanufacturing. *Comput. Ind. Eng.* **2023**, *184*, 109535. [CrossRef]
27. Liu, J.; Zhan, C.; Liu, Z.; Zheng, S.; Wang, H.; Meng, Z.; Xu, R. Equipment disassembly and maintenance in an uncertain environment based on a peafowl optimization algorithm. *Processes* **2023**, *11*, 2462. [CrossRef]
28. Gulivindala, A.K.; Bahubalendruni, M.R.; P, M.B.; Eswaran, M. Mechanical dis-assembly sequence planning for end-of-life products to maximize recyclability. *Sādhanā* **2023**, *48*, 109. [CrossRef]

29. Zhan, C.; Zhang, X.; Tian, G.; Pham, D.T.; Ivanov, M.; Aleksandrov, A.; Fu, C.; Zhang, J.; Wu, Z. Environment-oriented disassembly planning for end-of-life vehicle batteries based on an improved northern goshawk optimisation algorithm. *Environ. Sci. Pollut. Res.* **2023**, *30*, 47956–47971. [CrossRef] [PubMed]
30. Liao, H.Y.; Chen, Y.; Hu, B.; Behdad, S. Optimization-based disassembly sequence planning under uncertainty for human–robot collaboration. *J. Mech. Des.* **2023**, *145*, 022001. [CrossRef]
31. Qiu, L.; Dong, L.; Wang, Z.; Zhang, S.; Xu, P. Asynchronous parallel disassembly sequence planning method of complex products using discrete multi-objective optimization. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2022**, *236*, 1466–1482. [CrossRef]
32. Liu, J. *Research on the Equilibrium Problem of Sequentially Dependent Disassembly Line*; University of Electronic Science and Technology: Chengdu, China, 2018.
33. Yin, T.; Zhang, Z.; Jiang, J. A Pareto-discrete hummingbird algorithm for partial sequence-dependent disassembly line balancing problem considering tool requirements. *J. Manuf. Syst.* **2021**, *60*, 406–428. [CrossRef]
34. Pham, D.T.; Ghanbarzadeh, A.; Koç, E.; Otri, S.; Rahim, S.; Zaidi, M. The bees algorithm—A novel tool for complex optimisation problems. In *Intelligent Production Machines and Systems*; Elsevier Science Ltd.: Amsterdam, The Netherlands, 2006; pp. 454–459.
35. Hartono, N.; Ramírez, F.J.; Pham, D.T. Optimisation of robotic disassembly plans using the Bees Algorithm. *Robot. Comput.-Integr. Manuf.* **2022**, *78*, 102411. [CrossRef]
36. Mojtahedi, M.; Fathollahi-Fard, A.M.; Tavakkoli-Moghaddam, R.; Newton, S. Sustainable vehicle routing problem for coordinated solid waste management. *J. Ind. Inf. Integr.* **2021**, *23*, 100220. [CrossRef]
37. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
38. Shan, F.; Wu, Z.; Li, J. Research on the balance problem of lithium battery disassembly line based on improved particle swarm algorithm. *Intern. Combust. Engines Accessories* **2022**, *23*, 4–6.
39. Xu, Z.; Han, Y. Two sided disassembly line balancing problem with rest time of works: A constraint programming model and an improved NSGA II algorithm. *Expert Syst. Appl.* **2024**, *239*, 122323. [CrossRef]
40. Ji, J.; Wang, Y. Selective disassembly sequence optimization based on the improved immune algorithm. *Robot. Intell. Autom.* **2023**, *43*, 96–108. [CrossRef]
41. Fan, Y.; Zhan, C.; Aljuaid, M. Multi-Objective Disassembly Sequence Planning in Uncertain Industrial Settings: An Enhanced Water Wave Optimization Algorithm. *Processes* **2023**, *11*, 3057. [CrossRef]
42. Zhang, Q.; Li, H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
43. Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **1999**, *3*, 257–271. [CrossRef]
44. Fu, Y.; Zhang, Z.; Liang, P.; Tian, G.; Zhang, C. Integrated remanufacturing scheduling of disassembly, reprocessing and reassembly considering energy efficiency and stochasticity through group teaching optimization and simulation approaches. *Eng. Optim.* **2024**, 1–22. [CrossRef]