E-ISSN 2586-9027
Homepage : *https://tci-thaijo.org/index.php/SciTechAsia* Science & Technology Asia
Vol.30 No.4 October - December 2025 Page: [185-200]
Original research article

Solving Disassembly Line Balancing Problems with Fuzzy Parameters Using an Artificial Intelligence Technique

Arnat Watanasungsuit[1], Peerapop Jomtong[2], Choat Inthawongse[3], Chooosak Pornsing[4,*]

[1]*Hydrocarbon Solutions (Thailand) Co., Ltd. Thawi Wattana, Bangkok 10170, Thailand*
[2]*Department of Biomedical Engineering, Faculty of Health Sciences, Christian University, Nakhon Pathom 73000, Thailand*
[3]*Program in Smart Manufacturing Technology, Faculty of Industrial Technology, Muban Chom Bueng Rajabhat University, Ratchaburi 70150, Thailand*
[4]*Department of Industrial Engineering and Management, Faculty of Engineering and Industrial Technology, Silpakorn University, Nakhon Pathom 73000, Thailand*

Received 10 December 2024; Received in revised form 27 April 2025
Accepted 15 July 2025; Available online 17 December 2025

ABSTRACT

End-of-life (EOL) product recovery has become a critical issue driven by economic, social, and environmental concerns, along with stricter environmental regulations that emphasize disassembly and product recovery. Disassembly lines are used to dismantle EOL products into reusable components, but their efficiency depends heavily on accurately estimating task times, which are often uncertain. Since average task times cannot fully represent this uncertainty, task time can instead be modeled as a fuzzy number, allowing fuzzy logic to quantify representative values. This study introduces a disassembly line balancing problem where processing times are expressed as fuzzy numbers and solved using particle swarm optimization (PSO). The optimization aims to minimize the number of workstations, total idle time, maximum disassembly cost, and direction changes. The proposed method was benchmarked against LINGO and GUROBI solvers. Computational experiments, using the number of non-inferior solutions as a stopping criterion, demonstrated that the PSO-based approach achieved superior results. The findings indicate that the proposed method effectively outperforms existing algorithms, providing efficient and promising solutions for EOL product recovery and disassembly optimization.

Keywords: Disassembly line balancing; Fuzzy theory; Metaheuristics; Product recovery

***Corresponding author:** pornsing_c@su.ac.th
doi: 10.14456/scitechasia.2025.78

1. Introduction

The short product life cycle, emerging technological innovations, and dramatically increasing consumption contribute to colossal waste and environmental impacts. Additionally, legislation, social pressure, and economic attractiveness drive the requirement for recycling, remanufacturing, and recovery facilities. As a result, disassembly, which aims to extract reusable parts or subassemblies from end-of-life products or discarded products, is needed [1].

Disassembly is the methodical extraction of valuable parts and subassemblies from discarded products through operations [2]. The recycling process is a critical phase that impacts both environmental sustainability and the economic advantages for enterprises. Disassembly procedures conducted at a single workstation are inefficient and cannot be scaled to meet high demand [3]. Thus, the disassembly line is implemented as an effective method to attain scalability and automation in disassembly manufacturing [4, 5].

The disassembly line is the inverse process of the assembly line. Nonetheless, the disassembly line is somewhat more intricate than that. A significant degree of imbalance exists in the disassembly line. Consequently, the disassembly line balancing problem (DLBP) attracts much attention from manufacturing and operations researchers. There are numerous considerations related to disassembly lines [6-8]: product considerations, line considerations, part considerations, operational considerations, demand considerations, and assignment considerations. Hence, several works of literature exist in the last decade [9].

The disassembly process has a unique characteristic that is significantly different from the assembly process. The disassembly task time is not a deterministic number. Moreover, it cannot be represented by a statistical value such as average or median task time. As a result, the conventional disassembly line balancing design, which deals with deterministic data, cannot cope with real-world data that are imprecise, vague, and uncertain [10-12]. Fuzzy numbers may represent this uncertainty to reduce errors of imprecision.

To our knowledge, no particle swarm optimization has been used to solve such a complex problem. Multi-objective solutions with fuzzy disassembly task times disassembly line balancing problem (MFDLBP) are still challenging. The conceptual framework of this study is shown in a later section.

This study will make two main contributions: First, it proposes an improved metaheuristic algorithm for the fuzzy environment. Second, it presents the multi-objective optimization method for the disassembly line balancing problem and provides an application to solve the problem for sample products.

2. Materials and Methods

2.1 Problem statement and mathematical model

Merely considering the disassembly line balancing problem as the inverse of the assembly line balance problem needs to be revised. The disassembly line balance problem presents considerable challenges stemming from the substantial uncertainty associated with the quality and configuration of items on the line. The disassembly line exhibits certain parallels with the assembly line regarding its model formulation. However, prior studies on the assembly line balancing problem concentrate on two primary objectives: reducing the number of workstations and optimizing the workload distribution among workstations. This study examines the disassembly line balancing problem, which, in contrast to the assembly line balancing problem, incorporates two additional objectives: optimizing disassembly cost and the frequency of changes in disassembly direction.

The mathematical model of the issue is formulated based on the subsequent assumptions: the disassembly durations for each task are predetermined and regarded as constant; only one product is being disassembled, and it is being fully disassembled regardless of the recycling value of its parts; the cost of disassembling each task per unit time and the direction in which each part should be disassembled are known in advance. The disassembly line under examination employs a complete disassembly approach, with the quantity N of disassembly tasks, or components to be disassembled, being predetermined and known in advance.

2.2 Mathematical model

2.2.1 Notation

N Number of disassembly tasks

K Maximum possible number of workstations

k Workstation count $(1, 2, .., K)$

S_k Binary value, and when the workstation k is open, $S_k = 1$, otherwise, $S_k = 0$

n Disassembly task index

x_{ki} Binary value, 1 if disassembly task i is assigned to workstation k, else 0

$\grave{t}n$ Fuzzy disassembly time required for task n

$\grave{T}_C$ Fuzzy cycle time (maximum time available for each workstation)

y_{ij} If task i is executed before task j, $y_{ij} = 1$; otherwise, $y_{ij} = 0$. The priority matrix Y can be constructed, and $Y = [y_{ij}]_{N\times N}$

U_n Unit time disassembly cost of task n

r_n Disassembly direction of the n-th disassembly task in a given disassembly sequence

2.2.2 Objective functions

The optimization objective functions for the fuzzy disassembly line balancing problem are established based on the mathematical model of the problem found in the literature [13, 14]. The mathematical formulation of the MFDLBP is as follows:

Objective 1: minimize the number of workstations required:

$$\min f_1 = \sum_{k=1}^{K} S_k. \tag{2.1}$$

By determining the cycle time, the number of workstations in the disassembly production line can be significantly reduced, decreasing labor, equipment, and other requirements. Additionally, the disassembly cost will be reduced simultaneously.

Objective 2: minimize the total idle time of all the workstations while balancing the workloads among workstations:

$$\min f_2 = \sum_{k=1}^{K}(S_k \times \grave{T}_C - \sum_{n=1}^{N}(\grave{t}_n \times x_{kn}))^2. \tag{2.2}$$

This balance metric ensures the equitable workload distribution among various workstations on the disassembly line. Optimizing the balance between workstations and minimizing idle time is essential to enhancing work efficiency and fairness in disassembling production. It will effectively increase workstation usage.

Objective 3: minimize the sum of the maximum disassembly costs of all workstations:

$$\min f_3 = \sum_{k=1}^{K} S_k \times \grave{T}_C \times (\max[U_n | n \in (n | x_{kn} = 1)]. \quad (2.3)$$

The disassembly profit is calculated as the discrepancy between the total revenue obtained from the partial disassembly of the task and the expenses incurred from opening the workstations and performing the disassembly activities. The overall cost of disassembling the assembly line is calculated by adding the cost of each workstation. In this study, each workstation's unit time disassembly cost is the highest for internal disassembly chores within the workstation. To optimize the production benefit, minimizing the cost of disassembly is necessary.

Objective 4: minimize the disassembly direction change frequencies:

$$\min f_4 = \sum_{n=1}^{N} R_n, \quad R_n = \begin{cases} 1, & \text{if } r_n \neq r_{n-1} \\ 0, & \text{else} \end{cases}. \quad (2.4)$$

The direction measure was also created to quantify the performance of each solution sequence. Excessive changes in direction throughout the disassembly process will lead to more extended periods of invalid operation and, therefore, decrease disassembly efficiency. Therefore, minimizing the frequencies at which the disassembly direction changes is necessary.

2.2.3 Constraints

The total disassembly time required for any one workstation should not exceed the cycle time:

$$\sum_{n=1}^{N} (\grave{t}_n \times x_{kn}) \leq S_k \times \grave{T}_C, \ \forall k \in \{1, ..., K\}. \quad (2.5)$$

Each disassembly task must be allocated to a single workstation exclusively:

$$\sum_{k=1}^{K} x_{kn}, \ \ \forall n \in \{1, ..., N\}. \quad (2.6)$$

Every assignment must fulfill the precedence relationships required for the disassembly tasks:

$$\sum_{k=1}^{K} v \times x_{vi} \leq \sum_{w=1}^{K} w \times x_{wj}, \ \ \forall y_{ij} = 1. \quad (2.7)$$

2.3 Proposed approaching method

The solving method is based on particle swarm optimization (PSO), which has been modified to work with discrete search space. There are several discretization techniques; however, we will try using a new technique we published in the previous work: Sig-bo game discretization [15]. The proposed discrete particle swarm optimization with the Sig-bo game discretization technique is expressed in Table 1. Please note that we called the algorithm multi-objective optimization problem-particle swarm optimization (MOP-PSO).

2.4 Case experiments

This study is a series of our research. It starts with the disassembly line balancing problem with soft variety, for medium to large products, and with uncertainty. We

Table 1. MOP-PSO for DLBP.

Step	Description
1	Initialization (a) Set $k = 0$. (b) Randomly initialize the position of the particles in binary numbers. (c) Randomly initialize the velocity of the particles in binary numbers. For $i = 1, , m$ do $pbest_i^k = X_i$. Set $gbest^k = argmin_{i \in 1,,m} f(X_i)$.
2	Terminate Check. If the termination criteria hold stop. Then, the outcome of the algorithm will be .
3	For $i = 1, 2, , m$ do Calculating acceleration values. (a) Calculate $a_i^{k+1}(x_i^k(j))$. (b) Calculate $a_i^{k+1}(pbest_i^k(j))$. (c) Calculate $a_i^{k+1}(gbest^k(j))$. Updating particle swarm. (a) Update the velocity V_i^k using $v_i^{k+1}(j) = a_i^{k+1}(x_i^k(j)) + a_i^{k+1}(pbest_i^k(j)) + a_i^{k+1}(gbest^k(j))$ (b) Update the position X_i^k. $x_i^{k+1}(j) = \begin{cases} 1, & \text{if } 11 < v_i^{k+1}(j) \le 18 \\ 0, & v_i^{k+1}(j) \le 10 \\ \text{No Change} & \text{if } 10 < v_i^{k+1}(j) \le 11 \end{cases}$ Updating $pbest_i$ and $gbest$. (c) If $f(X_i) \le f(pbest_i)$ then $pbest = X_i$. (d) If $f(pbest_i) \le f(gbest)$ then $gbest = pbest_i$. End For.
4	Set $k = k + 1$.
5	Go to step 2.

have studied some physical products, such as the top-loaded washing machine and outdoor unit air conditioner (see Figs. 1-2). We will use these products as examples.

Fig. 1. Top-loaded washing machine.

Nonetheless, we need to model its structure differently from the previous study. We will use an AND/OR graph to represent the product structure. Furthermore, the fuzzy numbers are the disassembly time in form of (l,m,u). The details of the sample products data are shown in Tables 2-3.

Fig. 2. Outdoor unit air conditioner.

2.5 Performance measurement

2.5.1 Pareto optimal solutions

In prior literature, the multi-objective disassembly line problems are often addressed using linear weighting and sequential processing methods. The main aim of these two strategies is to convert a multi-objective problem into a single-objective problem, which is unable to harmonize several objective functions with conflicting relationships. The Pareto solution set considers the balance between several optimization objectives and provides answers that better reflect the actual problem scenario. It can be used as an approach to handle multi-objective problems.

The multi-objective disassembly line balancing problem can be solved by disassembly sequences that follow the disassembly precedence relation. The term ”feasible solution set” describes the set of all possible solutions. The dominance relationship between two feasible solutions, X_1 and X_2, is defined based on two requirements in Eq. (2.8) - (2.9).

$$f_d(X_1) \le f_d(X_2), \forall d \in \{1, 2, , D\}, \quad (2.8)$$

$$f_i(X_1) < f_i(X_2), \exists i \in \{1, 2, , D\}, \quad (2.9)$$

Table 2. Top-loaded washing machine disassembly tasks.

Task	Part	Time (sec.)	Unit cost (USD)	Direction	Precedence part
1	Bolts (2)	(3, 5, 7)	0.0040	L	-
2	Bolt (1)	(4, 7, 8)	0.0020	R	-
3	Cover	(2, 3, 5)	0.0575	L	-
4	Panel	(2, 3, 4)	1.0632	E	1, 2
5	Wash timer wiring	(20, 25, 28)	0.0655	L	4
6	Spin timer wiring	(32, 39 45)	0.0537	R	4
7	Washing timer knob	(2, 3, 5)	0.0060	L	5
8	Spin timer knob	(2, 3, 6)	0.0060	R	6
9	Timer switch bolts (2)	(30, 36, 38)	0.0040	R	4
10	Bind Tapping (5)	(82, 90, 101)	0.0200	L	4
11	Bind Tapping (5)	(165, 177, 182)	0.0200	R	4
12	Washing timer switch	(3, 5, 8)	0.0060	L	10, 11
13	Spin timer switch	(2, 3, 5)	0.0060	R	9
14	Washing selector knob	(3, 5, 8)	0.0060	L	13
15	Cycle selector knob	(2, 4, 6)	0.0060	R	12, 14
16	Buzzer	(3, 5, 6)	0.0155	R	10, 11
17	Pannel A	(1, 4, 6)	0.0632	E	12, 13
18	Switch cover	(2, 4, 7)	0.1050	L	17
19	Spinner lid	(2, 3, 5)	0.9955	R	17
20	Body b plate bolts (2)	(8, 10, 13)	0.0040	R	19
21	Nozzle holder bolts (2)	(2, 3, 5)	0.0040	L	18
22	Body bolts (3)	(3, 5, 7)	0.3300	L	3, 19
23	Nozzle holder	(3, 5, 6)	0.0560	L	21
24	Body plate a	(3, 5, 9)	1.1500	R	20
25	Body plate b	(20, 24, 26)	1.1500	R	22
26	Over flow filter a	(2, 3, 6)	0.0050	L	25
27	Special bolt	(13, 18, 20)	0.0020	L	3
28	Pulsator unit	(2, 3, 6)	0.1305	L	27
29	Back panel bolt (1)	(1, 3, 6)	0.0020	R	-
30	Back panel	(6, 11, 12)	1.0632	R	29
31	Back panel bolts (2)	(5, 9, 11)	0.0040	L	-
32	Back panel	(2, 3, 7)	1.0632	L	31
33	Base a bolt (3)	(1, 3, 5)	0.0060	R	-
34	Bolt	(295, 301, 318)	0.0020	R	30
35	Tub a	(25, 31, 39)	0.7350	E	30, 33
36	Drain tube	(5, 8, 10)	0.0350	L	30
37	Motor bolts (3)	(1, 4, 7)	0.0060	L	32, 35
38	Electric wire	(52, 62, 64)	1.4550	L	30
39	Motor	(1, 4, 6)	2.5600	L	35, 37

where D denotes the total number of objective functions, and f_d stands for the value of the associated objective solution.

Solution X_1 is considered non-dominated or non-inferior about X_2, meaning that X_1 dominates X_2 ($X_1 X_2$) and will be preserved while the nominated solutions are eliminated. A feasible solution X^* is deemed Pareto optimal or non-inferior if there exists no solution

Table 3. Outdoor unit air conditioner disassembly tasks.

Task	Part	Time (sec.)	Unit cost (USD)	Direction	Precedence part
1	Upper lid	(15, 19 20)	1.1300	E	-
2	Side cover	(1, 3, 5)	1.1300	L	1
3	Front cover	(20, 26, 28)	1.5956	L	1
4	Back cover	(6, 8, 9)	1.5956	E	1
5	Front grill	(16, 18, 20)	0.0155	L	3
6	Side grill	(5, 8, 9)	0.0155	L	3
7	Side frame (1)	(2, 3, 4)	0.0170	L	2, 3
8	Side frame (1)	(12, 14, 18)	0.0170	L	2, 3
9	Side frame (1)	(2, 3, 4)	0.0170	R	7, 8
10	Side frame (2)	(25, 27, 28)	0.7927	R	7, 8
11	Side frame (2)	(2, 3, 4)	0.7927	R	7, 8
12	Motor and fan case	(4, 6, 9)	0.9550	L	10, 11
13	Wiring harness of motor	(2, 3, 4)	0.0559	E	12
14	Fan	(2, 5, 6)	2.4054	L	13
15	Fan base (1)	(10, 13, 14)	0.9397	E	14
16	Fan base (1)	(26, 29, 30)	0.9397	E	14, 15
17	Heating coil	(8, 10, 11)	2.3270	R	-
18	Wiring harness of fan	(12, 14, 15)	0.0559	E	14, 17
19	Electric circuit of heating coil	(1, 2, 3)	0.9955	R	18
20	Circuit box	(25, 28, 30)	0.0655	E	18
21	Heating coil and compressor frame (2)	(14, 16, 17)	0.2308	R	17, 20
22	Heating coil and compressor frame (2)	(2, 5, 6)	0.2308	L	17, 20
23	Compressor frame	(35, 38, 41)	0.1150	E	10, 11, 12, 13
24	Compressor	(7, 8, 9)	2.6216	R	23
25	Compressor base	(4, 5, 7)	0.3321	E	24
26	Heating coil base	(20, 22, 26)	0.3321	R	22, 24

sequence X such that X is strictly preferred to X^*. The assemblage of all Pareto optimal solutions is termed the Pareto optimal solution set, whereas the aggregation of objective function values associated with the disassembly sequence is designated as the Pareto optimal frontier.

Fig. 3 depicts the potential solutions of a two-objective optimization problem. According to the data presented in Fig. 3, solution B outperforms solution E regarding both objectives. Therefore, solution E is considered inferior to solution B. Solution B outperforms solution C in objective f_1 but is subordinate to solution C in objective f_2, leading to no substantial distinction between solution B and C. The solutions A, B, C, and D provide Pareto optimal solutions for the two-objective optimization problem. In the context of decision-making, non-dominated solutions are considered as the threshold values for the objective functions. These solutions always lie on the boundary of the decision space, known as the Pareto optimum front in multi-objective optimization. The hollow points E, F, and G represent potential solutions within the decision space; however, they are situated on the best frontier and will be either directly or indirectly surpassed by the Pareto optimal solutions found in the ideal frontier.

The current Pareto optimum solutions are stored in an external file called Q. During each iteration of the algorithm, every new solution p_i from the population P will undergo comparison with each non-inferior solution q_j stored in Q. If p_i is

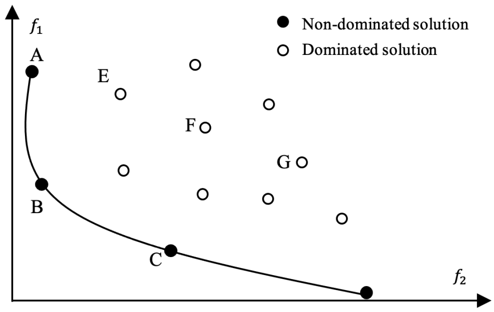

Fig. 3. Pareto optimal solutions.

less than q_j, the new solution p_i will be recorded in Q, and the solution q_j will be deleted. If q_j is less than p_i, the Q will remain unaltered. If p_i and q_j are mutually exclusive, they are placed in Q. Subsequently, the Q will systematically change to achieve the most favorable non-inferior solutions.

2.5.2 Coefficient of variation

The coefficient of variation (CV) is a statistical metric that quantifies the degree of dispersion of data points within a series relative to the mean. The coefficient of variation is determined by dividing the standard deviation by the mean and is typically represented as a percentage. A larger coefficient of variation (CV) indicates a greater degree of dispersion around the mean.

$$\hat{C}_v = \frac{s}{\bar{x}}, \tag{2.10}$$

where $\hat{C}_v$ is the coefficient of variation, s is the sample standard deviation, and $\bar{x}$ is the sample mean.

2.6 Parameter setting

A drawback of the metaheuristic approach is that many parameters need to be predetermined. In this study, we will cultivate the previous study by using their initial parameter settings in our numerical experiment.

Table 4 shows the details of the proposed method's parameter settings. These settings do not guarantee the best solution; however, they are good enough based on previous studies [15, 16].

Table 4. Parameter setting.

Parameter	Description	Value
ω_{max}	Maximal inertia weight	0.7
ω_{min}	Minimal inertia weight	0.3
$\Delta\omega$	Inertia weight step-size	0.1
ϕ_1, ϕ_2	Acceleration constant	2.0
N	Population size	100
p	Number of sub-swarms	4
δ	Bounce-factor	0.5
T	Iteration number	1500

2.7 Benchmark approaches

We used the LINGO package to solve the mixed-integer linear programming. The mathematical model was coded in LINGO. Furthermore, we also used GUROBI on the NEOS Server to solve this problem by coding the mathematical model into AMPL. The sites of these two solving tools are https://www.lindo.com/index.php and https://neos-server.org/neos/.

Technically, LINGO (Lindo Global Optimization) is a powerful optimization software tool commonly used for solving linear, nonlinear, integer, and mixed-integer optimization problems. It provides a user-friendly interface for modeling and solving optimization problems in various logistics, manufacturing, finance, and telecommunications industries. LINGO offers a comprehensive set of features, including:

1. Modeling language: LINGO provides a high-level modeling language that allows users to express optimization problems naturally and intuitively using mathematical notation.

2. Solver engine: LINGO uses advanced optimization algorithms to solve

optimization problems efficiently. It includes algorithms for linear programming (LP), quadratic programming (QP), mixed-integer programming (MIP), nonlinear programming (NLP), and global optimization.

3. Interactive environment: LINGO offers an interactive environment where users can easily build, solve, and analyze optimization models. It provides tools for visualizing results, debugging models, and analyzing sensitivity.

4. Integration: LINGO can be integrated with other software tools and programming languages such as Excel, MATLAB, and Python, allowing users to leverage its optimization capabilities within their existing workflows.

Overall, LINGO is a versatile optimization tool that enables users to tackle complex optimization problems and find optimal solutions efficiently.

Besides, GUROBI is not an algorithm per se but a powerful optimization solver software package. It is designed to solve various types of mathematical optimization problems, including linear programming (LP), quadratic programming (QP), mixed-integer linear programming (MILP), mixed-integer quadratic programming (MIQP), and more.

The software implements state-of-the-art optimization algorithms, including primal and dual simplex methods, interior-point methods, and branch-and-bound techniques for mixed-integer problems. GUROBI is known for its efficiency, robustness, and ability to handle large-scale optimization problems efficiently.

It's widely used in academia and industry for applications such as operations research, supply chain management, finance, engineering, etc. GUROBI provides APIs for various programming languages such as Python, C, C++, Java, and MATLAB, making it accessible and easy to integrate into existing software systems.

3. Result and Discussions

3.1 Computational results

The top-loaded washing machine has 39 disassembly tasks while the outdoor unit air conditioner has 26. The fuzzy cycle time $\acute{T}_c$ for the top-loaded washing machine is equal to (300,400,500) while the $\acute{T}_c$ for the outdoor unit air conditioner is equal to (150,200,300).

Table 5 shows the results of thirty independent executions on a personal computer with an Intel Core i7-8750 CPU@2.2 GHz processor, 8.0 GB RAM, and MS Windows 10. Please note that we did not compare the computational time of the approaches because the GUROBI algorithm was run on a cloud server.

Table 5. The number of non-inferior solutions.

Product	Statistic	MOP-PSO	LINGO	GUROBI
		-1	-2	-3
Top-loaded washing machine	Max	14	8	15
	Min	8	1	3
	Average	10.5	4.75	7.7
	SD.	2.04	2.38	3.54
		0.19	0.5	0.46
Outdoor unit air conditioner	Max	16	8	12
	Min	7	1	4
	Average	9.5	5.1	9.5
	SD.	2.84	2.25	1.76
		0.3	0.44	0.19

From Table 5, MOP-PSO looks outstanding among the competitive algorithms. It could find more non-inferior solutions; additionally, the coefficients of variation were lower than others on both sample products. The less $\hat{C}_v$ represents a more consistent approach.

However, we could not prove that these three algorithms significantly differed in statistical analysis perspectives. Accordingly, we needed to conduct a deep analysis to confirm that the proposed method outperforms other competitive methods.

First, the boxplot diagram was portrayed for each sample product. Fig. 4 shows the boxplot of the approaches on the top-loaded washing machine, in which 1, 2, and 3 means Modified PSO, LINGO, and GUROBI, respectively.

Fig. 4 shows that LINGO was different from the group. However, MOP-PSO and GUROBI do not appear to be significantly different.

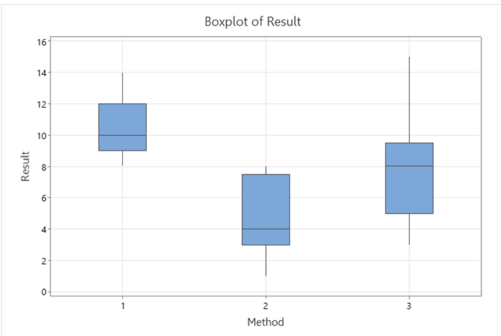

Fig. 4. Boxplot of non-inferior solution number of the top-loaded washing machine.

Thus, we need to carry out the analysis of variance (ANOVA) based on the hypothesis below. H_O: All approaches are not different performance. H_1: There is one approach different from others. Figs. 5-8 are the analysis results from an analysis tool for statistical data. Fig. 5 shows that the null

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Method	2	330.7	165.350	22.18	0.000
Error	57	424.9	7.455		
Total	59	755.6			

Fig. 5. ANOVA table for the top-loaded washing machine.

hypothesis is rejected. This means that at least one method is different from others. The adjusted R-squared which tell us how well a model fits data and predicts the data

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
2.73043	43.76%	41.79%	37.69%

Fig. 6. Model adequacy for the top-loaded washing machine.

Coefficients

Term	Coef	SE Coef	T-Value	P-Value	VIF
Constant	7.650	0.352	21.70	0.000	
Method					
1	2.850	0.499	5.72	0.000	1.33
2	-2.900	0.499	-5.82	0.000	1.33

Fig. 7. Model coefficients for the top-loaded washing machine.

Fits and Diagnostics for Unusual Observations

Obs	Result	Fit	Resid	Std Resid
46	14.000	7.700	6.300	2.37 R
49	15.000	7.700	7.300	2.74 R

Fig. 8. Unusual data for the top-loaded washing machine.

is 41.79% which is looking good, see Fig. 6.

Fig. 7 explains the coefficients of the model which implies that all independent variables are statistically significant to the model. Finally, the unusual data were pointed out by the tool in Fig. 8.

The next question is how different they are? We have known only that there is at least one method different from others but we do not know the detail yet. Accordingly, we deployed the Tukey test with 95% confidence level to cluster the method. Fig. 9 illustrates the Tukey test result.

Grouping Information Using the Tukey Method and 95% Confidence

Method	N	Mean	Grouping		
1	20	10.50	A		
3	20	7.70		B	
2	20	4.75			C

Means that do not share a letter are significantly different.

Fig. 9. Tukey test for the top-loaded washing machine.

Fig. 9 shows that the three groups that are significantly different from each other. Group A yields the highest mean, which is the MOP-POS method. Group B yields the second rank of mean, which is the GUROBI method. The last one is Group C, which is the LINGO method.

We then analyzed the result from the air conditioner outdoor unit product. First, a boxplot diagram was portrayed for each sample product. Fig. 10 shows the boxplot of the approaches on the air conditioner outdoor unit, in which 1, 2, and 3 means Modified PSO, LINGO, and GUROBI, respectively.

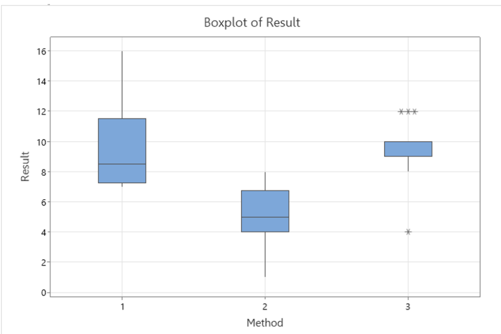

Fig. 10. Boxplot of non-inferior solution number of outdoor unit air conditioner.

Fig. 10 shows that LINGO was different for the group. However, MOP-PSO and GUROBI do not appear to be significantly different. Thus, we need to carry out the analysis of variance (ANOVA) based on the hypothesis we declared earlier. Figs. 11-14 are the analysis results from a statistical data analysis tool.

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Method	2	258.1	129.067	23.90	0.000
Error	57	307.8	5.400		
Total	59	565.9			

Fig. 11. ANOVA table for the outdoor unit air conditioner.

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
2.32379	45.61%	43.70%	39.74%

Fig. 12. Model adequacy for the outdoor unit air conditioner.

Coefficients

Term	Coef	SE Coef	T-Value	P-Value	VIF
Constant	8.033	0.300	26.78	0.000	
Method					
1	1.467	0.424	3.46	0.001	1.33
2	-2.933	0.424	-6.91	0.000	1.33

Fig. 13. Model coefficients for the outdoor unit air conditioner.

Fits and Diagnostics for Unusual Observations

Obs	Result	Fit	Resid	Std Resid	
5	16.000	9.500	6.500	2.87	R
6	16.000	9.500	6.500	2.87	R
56	4.000	9.500	-5.500	-2.43	R

Fig. 14. Unusual data for the outdoor unit air conditioner.

Fig. 11 shows that the null hypothesis is rejected. This means that at least one method is different from others. The adjusted R-squared, which tells us how well a model fits data and predicts the data, is 43.70%, which looks good (see Fig. 12).

Fig. 13 explains the coefficients of the model, which implies that all independent variables are statistically significant to the model. Finally, the tool in Fig. 14 pointed out the unusual data.

We also deployed the Tukey test with a 95% confidence level to cluster the method. Fig. 15 illustrates the Tukey test result.

Fig. 15 shows that the two groups are significantly different from each other. Group A yields the highest mean, as do the MOP-POS and GUROBI methods. Group B yields the lower mean, which is the LINGO method. Obviously, the proposed method was not different from the GUROBI package on the outdoor unit air conditioner.

Grouping Information Using the Tukey Method and 95% Confidence

Method	N	Mean	Grouping	
1	20	9.5	A	
3	20	9.5	A	
2	20	5.1		B

Means that do not share a letter are significantly different.

Fig. 15. Tukey test for the outdoor unit air conditioner.

Unfortunately, we could not test their efficiencies, such as convergence rate and computational time, because the GUROBI method was run on a cloud server.

3.2 Explicit computational results

Then, the explicit result analysis was conducted. The stopping criterion was changed to force the algorithms to find the non-inferior solutions at fifteen to stop the execution. Many times, the solution found so far may be identical. We must accept the identical solution of up to two solutions. Furthermore, we are interested in comparing the explicit results of three algorithms. Legitimately, specific results could not guarantee their performance in general. However, they could guide a practitioner to consider a solving tool for his on-hand problem.

Tables 5-6 show the computational results of the top-loaded washing machine and the outdoor unit air conditioner, respectively. The bold numbers are the best solutions found so far.

The solutions in Tables 5 and 6 illustrate the proposed algorithm's performance. The best solutions of the proposed algorithm outperformed the competitive algorithms in both sample products. It is interesting to note the significant differences in the results. Figs. 16-18 draw the descriptive statistics of three algorithms with maximum, minimum, and average values for objective functions 2 to 4.

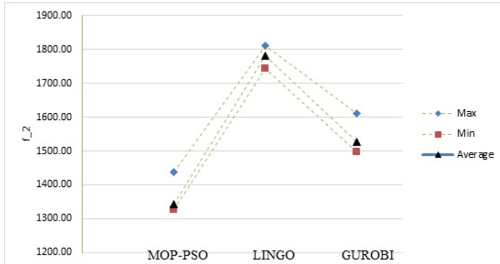

Fig. 16. Descriptive statistics comparison of f_2 on the top-loaded washing machine.

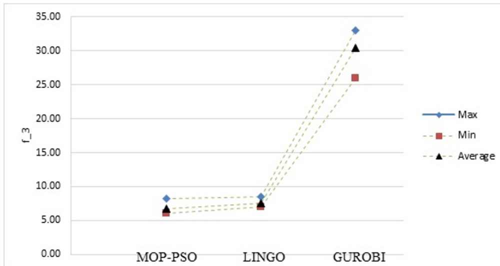

Fig. 17. Descriptive statistics comparison of f_3 on the top-loaded washing machine.

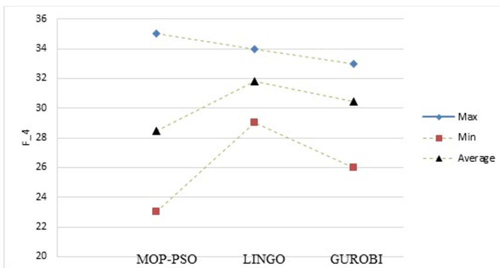

Fig. 18. Descriptive statistics comparison of f_4 on the top-loaded washing machine.

Figs. 19-21 show the descriptive statistics of three algorithms with maximum, minimum, and average values for objective functions 2 to 4.

4. Conclusion and Outlook

This research has two primary aims. Initially, we introduce an enhanced metaheuristic algorithm aimed at addressing

Table 6. Explicit computational results on the top-loaded washing machine.

No.	MOP-PSO				LINGO				GUROBI			
	f_1	f_2	f_3	f_4	f_1	f_2	f_3	f_4	f_1	f_2	f_3	f_4
1	6	1330.65	7.251	25	6	**1742.65**	8.354	32	6	**1498.25**	6.450	32
2	6	1330.65	6.105	26	6	1785.32	7.144	32	6	1522.03	7.190	27
3	6	1330.00	8.251	32	6	1773.43	7.256	33	6	1532.17	6.450	30
4	6	1341.28	6.251	26	6	1803.55	**7.015**	34	6	1518.41	7.352	32
5	6	1330.65	6.250	26	6	1780.08	7.946	**30**	6	1511.68	7.215	32
6	6	1341.28	6.433	23	6	1785.32	8.490	32	6	1499.06	8.235	26
7	6	1330.65	8.251	32	6	1793.57	7.013	31	6	**1498.25**	7.853	32
8	6	1438.02	6.251	25	6	1810.22	8.456	**30**	6	1524.04	6.441	28
9	6	1330.65	6.251	32	6	1785.32	7.144	31	6	1511.68	6.441	32
10	6	1345.02	6.055	35	6	1769.86	7.235	33	6	1532.17	**6.402**	33
11	6	1330.65	6.251	26	6	1785.32	8.354	29	6	1532.17	6.508	32
12	6	1345.02	6.203	33	6	**1742.65**	7.144	33	6	1511.68	6.441	32
13	6	1346.51	6.080	**24**	6	1769.05	7.100	32	6	1610.31	**6.402**	32
14	6	1341.28	**6.055**	28	6	1803.55	7.069	33	6	1509.74	6.501	30
15	6	**1326.25**	8.256	34	6	1785.32	8.435	32	6	1576.09	7.233	27
Max	6	1438.02	8.256	35	6	1810.22	8.490	34	6	1610.31	8.235	33
Min	6	1326.25	**6.055**	23	6	1742.65	7.013	29	6	1498.25	6.402	26
Average	6	1342.57	6.680	28.47	6	1781.01	7.610	31.80	6	1525.85	6.874	30.47

Table 7. Explicit computational results on the outdoor unit air condition.

No.	MOP-PSO				LINGO				GUROBI			
	f_1	f_2	f_3	f_4	f_1	f_2	f_3	f_4	f_1	f_2	f_3	f_4
1	5	713.25	4.371	21	5	784.78	6.945	24	5	743.22	6.782	20
2	5	728.04	**4.145**	20	5	845.17	6.925	24	5	731.09	7.895	21
3	5	713.05	5.435	20	5	765.22	7.093	21	5	725.55	7.999	21
4	5	700.32	5.311	22	5	832.08	8.221	**21**	5	732.09	6.452	21
5	5	709.05	6.032	21	5	785.32	7.003	22	5	728.44	6.923	22
6	5	**698.48**	6.622	19	5	813.06	7.052	23	5	731.09	8.909	20
7	5	715.19	7.735	**17**	5	793.03	**6.832**	25	5	745.11	7.839	19
8	5	728.04	4.260	19	5	814.06	7.003	**21**	5	812.15	**6.210**	22
9	5	712.13	6.429	18	5	765.22	9.890	**21**	5	798.46	6.955	20
10	5	726.54	4.995	21	5	830.11	8.455	**21**	5	734.52	6.350	21
11	5	715.19	4.490	20	5	760.22	8.725	22	5	795.34	6.987	19
12	5	701.16	6.063	20	5	829.25	7.000	22	5	**718.52**	8.352	23
13	5	699.56	6.599	22	5	747.65	9.946	24	5	732.09	7.895	20
14	5	713.25	5.073	19	5	769.32	6.999	25	5	720.45	7.022	22
15	5	721.11	5.002	20	5	800.55	7.000	23	5	735.12	7.052	20
Max	5	728.04	7.735	22	5	832.08	9.946	25	5	812.15	8.909	23
Min	5	698.48	4.145	**17**	5	**747.65**	6.832	21	5	718.52	6.210	19
Average	5	712.96	5.504	19.93	5	792.05	7.852	22.33	5	745.72	7.346	20.79

DLBP in the context of uncertainty. Secondly, we assess the effectiveness of the proposed method by implementing it in the design of a disassembly line balancing for sample products.

The modified metaheuristics in this study was a multi-objective particle swarm optimization (MOP-PSO), which deploys

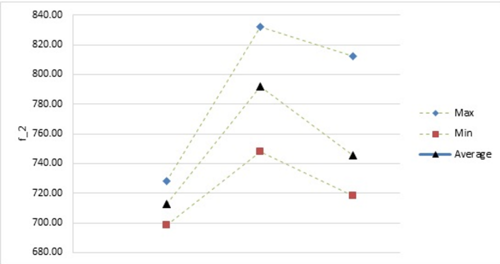

Fig. 19. Descriptive statistics comparison of f_2 on the outdoor unit air conditioner.

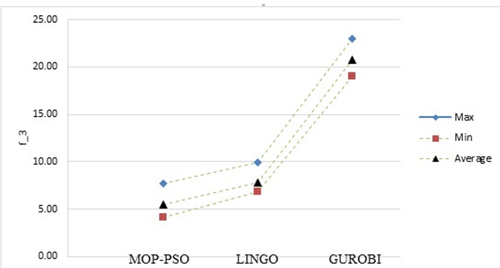

Fig. 20. Descriptive statistics comparison of f_3 on the outdoor unit air conditioner.

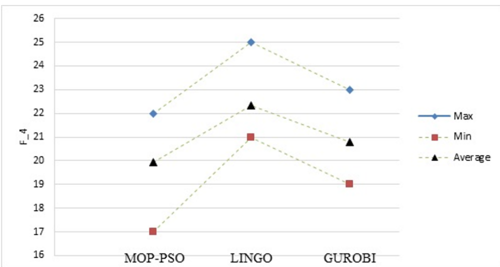

Fig. 21. Descriptive statistics comparison of f_4 on the outdoor unit air conditioner.

the Sic-bo game in its discretization mechanism and makes the elite list using the Pareto-optimality principle. The algorithm was synchronized to four objectives in the mathematical model of DLBP: disassembly line balancing problem, minimizing the number of workstations, minimizing total idle time, minimizing maximum disassembly cost, and minimizing total direction changes. The complexity of this new method was that it could deal with fuzzy disassembly task time. The fuzzy numbers were operated before the algorithm process. Then, the defuzzification was executed to give the solutions.

The proposed method was tested by comparing it to competitive solvers like LINGO and GUROBI. Two sample products were used: the top-loaded washing machine, which required thirty-nine disassembly tasks, and the outdoor unit air conditioner, which required twenty-six disassembly tasks. Please note that this study did not consider the destructive disassembly process as the research attribute.

The computational results showed that the proposed method outperformed the number of non-inferior solutions found. Furthermore, it promised consistency by yielding the lowest coefficient of variation numbers. Explicit computational experiments were carried out. This step changed the stopping criterion to the number of non-inferior solutions found. The results showed that the proposed approach surpassed the competitive algorithms in this study. It provided promising solutions for both sample products.

However, this study did not cover many circumstances in the real world. The quality of the disassembled component can be modeled as a fuzzy number. The determination between destructive disassembly and non-destructive disassembly can be solved and optimized to support the determination of disassembly line design.

Moreover, the disassembly line design layout can be included in the problem; it can be modeled as a mathematical model and solved by metaheuristics in the context of a multi-objective optimization problem.

Acknowledgments

This research project was granted by the Silpakorn University Research, Innovation and Creative Fund. We express our gratitude to the anonymous reviewers for their meticulous examination of our work and their numerous astute observations and recommendations.

References

[1] Ding LP, Feng YX, Tan JR, Gao YC. A new multi-objective ant colony algorithm for solving the disassembly line balancing problem. The International Journal of Advanced Manufacturing Technology. 2010 May;48:761-71.

[2] Avikal S, Jain R, Mishra PK. A Kano model, AHP and M-TOPSIS method-based technique for disassembly line balancing under fuzzy environment. Applied Soft Computing. 2014 Dec 1;25:519-29.

[3] Zhang X, Tian G, Fathollahi-Fard AM, Pham DT, Li Z, Pu Y, Zhang T. A chance-constraint programming approach for a disassembly line balancing problem under uncertainty. Journal of Manufacturing Systems. 2024 Jun 1;74:346-66.

[4] Yin T, Zhang Z, Jiang J. A Pareto-discrete hummingbird algorithm for partial sequence-dependent disassembly line balancing problem considering tool requirements. Journal of Manufacturing Systems. 2021 Jul 1;60:406-28.

[5] Wang K, Li X, Gao L, Li P, Gupta SM. A genetic simulated annealing algorithm for parallel partial disassembly line balancing problem. Applied Soft Computing. 2021 Aug 1;107:107404.

[6] Edis EB, Edis RS, Ilgin MA. Mixed integer programming approaches to partial disassembly line balancing and sequencing problem. Computers & Operations Research. 2022 Feb 1;138:105559.

[7] Budak A. Sustainable reverse logistics optimization with triple bottom line approach: An integration of disassembly line balancing. Journal of Cleaner Production. 2020 Oct 10;270:122475.

[8] Çil ZA, Mete S, Serin F. Robotic disassembly line balancing problem: A mathematical model and ant colony optimization approach. Applied Mathematical Modelling. 2020 Oct 1;86:335-48.

[9] Xu W, Cui J, Liu B, Liu J, Yao B, Zhou Z. Human-robot collaborative disassembly line balancing considering the safe strategy in remanufacturing. Journal of Cleaner Production. 2021 Nov 15;324:129158.

[10] Wang K, Li X, Gao L, Garg A. Partial disassembly line balancing for energy consumption and profit under uncertainty. Robotics and Computer-Integrated Manufacturing. 2019 Oct 1;59:235-51.

[11] Paprocka I, Skołud B. A predictive approach for disassembly line balancing problems. Sensors. 2022 May 22;22(10):3920.

[12] Zhang L, Zhao X, Ke Q, Dong W, Zhong Y. Disassembly line balancing optimization method for high efficiency and low carbon emission. International Journal of Precision Engineering and Manufacturing-Green Technology. 2021 Jan;8:233-47.

[13] *Ö*zceylan E, Kalayci CB, Güngör A, Gupta SM. Disassembly line balancing problem: a review of the state of the art and future directions. International Journal of Production Research. 2019 Aug 29;57(15-16):4805-27.

[14] Deniz N, Ozcelik F. An extended review on disassembly line balancing with bibliometric & social network and future study realization analysis. Journal of Cleaner Production. 2019 Jul 10;225:697-715.

[15] Pornsing C, Sangkhiew N, Sakonwittayanon P, Jomtong P, Ohmori S. A new discretization technique for enhancing discrete particle swarm optimization's performance. Science & Technology Asia. 2022 Sep 28:204-15.

[16] Pornsing C, Sodhi MS, Lamond BF. Novel self-adaptive particle swarm optimization methods. Soft Computing. 2016 Sep;20:3579-93.