E-ISSN 2586-9027

Homepage : *https://tci-thaijo.org/index.php/SciTechAsia* Science & Technology Asia

Vol.30 No.4 October - December 2025 Page: [303-322]

Original research article

Disassembly Line Balancing Design for Mixed-Model Disassembly Process Using a Modified Metaheuristic Approach

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 11 February 2025; Received in revised form 8 June 2025
Accepted 4 July 2025; Available online 17 December 2025

ABSTRACT

This study has two main objectives: to propose a mathematical model for the mixed-model disassembly line balancing problem and to develop a customized solving technique. The model aims to minimize the disassembly line length and the number of opened stations while maximizing workload smoothness in a two-sided disassembly line. The solver, based on the particle swarm optimization (PSO) algorithm, was enhanced through a new discretization method and the survival sub-swarm PSO strategy, enabling it to handle multi-objective optimization via Pareto optimality for constructing the elite list. To validate the approach, experiments were conducted on a top-loaded washing machine with different takt times (71, 80, 90, and 100 seconds). Four competitive algorithms—NSGA-II, SPEA2, BARON, and MINLP—were used for comparison. Performance was evaluated using three indicators: inverted generational distance (IGD), hypervolume (HV), and ratio (R). The results showed that the proposed method consistently outperformed the other algorithms, achieving superior accuracy, efficiency, and stability in delivering optimal and reliable solutions.

Keywords: Disassembly line; Mixed-model; Metaheuristics; Particle swarm optimization; Multi objective

*Corresponding author: pornsing_c@su.ac.th

doi: 10.14456/scitechasia.2025.86

1. Introduction

The primary goal of product recovery is to minimize the amount of trash disposed of in landfills or incinerators. The responsibility to address environmental concerns has become a requirement enforced by governmental rules and influenced by societal expectations [1]. Recycling and remanufacturing are employed to recover a product that has become outdated or reached its end-of-life (EOL) stage. Recycling is a process that involves disassembling, sorting, and applying chemical processes to end-of-life products in order to recover their material content. In contrast, remanufacturing is a procedure that seeks to preserve the product's initial characteristics and involves disassembly, categorization, restoration, and reassembly to achieve a desired level of excellence [2]. Disassembly has proven efficient in recovering materials and products in various product recovery processes. The process involves deliberately separating specific components, sub-assemblies, and materials [3].

The disassembly procedures can also be categorized into four sub-problems. The process encompasses various stages: disassembly planning, scheduling, sequencing, and disassembly line balancing (DLB). DLB involves allocating disassembly jobs to various workstations to meet all the required jobs sequences while optimizing particular criteria [4]. The DLB is frequently linked with assembly line balancing (ALB). However, it exhibits greater complexity in terms of operating aspects.

Numerous research has put forth the challenges associated with disassembly line balancing. Reference [5] sought to decrease the number of open workstations while concurrently optimizing the allocated tasks and maximizing the efficiency of the disassembly line's arrangement and material handling equipment. Reference [6] sought to optimize the revenue obtained from a disassembly process. The likelihood of redistributing the remaining tasks in case of a task failure was evaluated. The reader is directed to consult the publication authored by [7].

Based on the author's extensive understanding, the availability of disassembly line balance combined with mixed-model line design is limited. Reference [8] reported that most studies on disassembly line balancing in the literature between 1999 and 2020 focused on the single-model approach. Ninety-six percent of all published publications in this particular field of inquiry are attributed to it. According to reports, the design characteristics of mixed-model disassembly lines were identified in a few articles. Notably, the product assortment within reverse logistics is not without flaws. The acquisition of the obsolete product in its entirety is not feasible. Typically, products arrive at the recycling center through many means. The optimal approach is to categorize them into different types of soft product variety. Hence, the rationale behind implementing a disassembly line that accommodates mixed-model goods is justifiable. Consequently, the work addresses the knowledge gap in investigating mixed-model disassembly line balance. It is imperative to explore this topic from several viewpoints in order to enhance our understanding.

2. Mixed-Model Disassembly Line

2.1 Disassembly process

Disassembly is the methodical process of breaking down a unit into its constituent parts, components, or other groups [9]. Selective separation is a crucial step in the recovery of materials and products since it enables the specific extraction of desired components and substances. The disassembly line predates the assembly line by approximately one century. One example of a 19th-century slaughterhouse is found in Chicago. The procedures employed in these slaughterhouses are widely acknowledged as the pioneering implementation of production lines. The Ford Motor Company in the early 1900s is often credited with pioneering the first efficient assembly lines.

The efficiency of assembly lines experienced significant advancements during the course of the following century. In contrast, research on disassembly lines has only recently commenced. The success of assembly lines necessitates the disposal of a large number of end-of-life products, which are perceived to have a negative environmental impact. The innovative concept suggests that the most effective approach to studying assembly problems is by treating them as reverse disassembly problems. The subsequent surge in disassembly research has consistently grown due to ongoing motivation from government regulation, corporate profitability, and customer demands. Ultimately, the disassembly line was given its initial distinct characterization in the 2000s. Since then, mathematical models and solution approaches have been explored. The following sub-sections are the disassembly line considerations.

2.1.1 Product considerations

A disassembly line has the capability to handle a group of related products. A product family refers to a single product that maintains the same initial configuration across all acquired products. For instance, this could include just personal computers with certain specs. Nevertheless, the differentiated products exhibit only marginal variations from one another. Therefore, the disassembly line may receive multiple partially disassembled items and subassemblies that have significantly or completely different configurations, such as personal computers, printers, digital cameras, and so on.

2.1.2 Line considerations

Based on assembly-line layouts, there are many configurations such as serial, parallel, circular, U-shape, cellular, and two-sided lines. All of them are designed and applied to enable more efficient disassembly lines. There are also paced and unpaced lines. The paced line may be used to control the flow of products. It mitigates work in process, required space, and bottleneck workstations. However, sometimes, the task timings exhibit excessive variability. In this situation, the unpaced line may be appropriate.

2.1.3 Part considerations

The part considerations pertain to the quality of the components. A faulty part refers to a component that deviates from its original structure and/or functioning standards. There are two main categories of faults: physical defects and functional defects. The physical flaw results in a deviation from the original design's geometric parameters. A functional fault refers to a situation when a part does not perform its intended function as originally planned.

2.1.4 Operational considerations

The operational concerns encompass seven aspects: variability of disassembly job timeframes, the early departure of workpieces, self-omission of workpieces, omission of workpieces, the disappearance of workpieces, revisiting of workpieces, and the explosion of work. One prevalent issue is the unpredictability of job times during disassembly. The variation in question is contingent upon many aspects that are linked to the caliber of the products and the condition of the disassembly workstation. The early departure of workpieces refers to a situation where a present workstation is unable to complete the assigned disassembly assignment due to a defect. The workpiece has the potential to depart from the workstation prematurely. The concept of self-skipping workpieces refers to the situation where certain activities allocated to a workstation are not carried out owing to defects or precedence connections. The workpiece departs from the workstation prematurely without undergoing any processing.

The skipping work-pieces consideration means the work-piece is not designed to be disassembled in the current workstation. Thus, it can skip the current workstation. The disappearing work-pieces consideration means a defective part may be taken off before the assigned workstation. Accordingly, when the part comes to the assigned workstation, it may skip the task. The concept of revisiting workpieces refers to the possibility of a workpiece now located at workstation j returning to a previous workstation $(j - p)$, where $(j - p)$ is greater than or equal to 1 and p is a positive integer. The concept of exploding workpieces refers to the phenomenon where a single workpiece may fracture or separate into multiple pieces while transported along the disassembly line.

2.1.5 Demand considerations

The need for pre-owned components is a significant subject in disassembly line issues. While there is a definite demand for addressing assembly line problems, the market for addressing disassembly line problems remains uncertain. The demand for disassembly lines can be categorized into three scenarios: demand for a single part, demand for numerous parts, and demand for all components. Moreover, the necessary components are occasionally found in several formats. Within the realm of literature, there exist three distinct categories of demand. In the first category, the demand source has the option to accept a component without any modifications. For instance, it is conceivable that a component may require its own material composition. In the second category, the demand source exclusively takes parts of high quality. Defective parts are not accepted. The third category of demand encompasses both the geometric and functional aspects of the demand source. This is more serious in the disassembly process. The receiving parts must pass the criteria of dimension and function of those parts.

2.1.6 Assignment considerations

There are restrictions on unique machining and tooling either at specific workstations or at a limited number of workstations. Though, products have the precedence relationships. Accordingly, the task assignment on a specific workstation is inevitable. This factor affects repositioning activities, revisit processes, tool changes in the disassembly process, etc. Additional circumstances exist that are relevant to assignment decisions, including the dismantling of dangerous components. In order to mitigate the risk of contaminating the remainder of the process, it is imperative to allocate these components to specialized workstations. For example, the disassembly processes of car batteries need to separate between normal disassembly processes with battery containers and the processes related to electrodes inside the battery.

2.2 Mixed-model disassembly process

Due to imperfect matching between the demand and supply of disassembly processes, the design of the disassembly process encounters the issue of several product disassembly lines. The prevalent mixed-model disassembly lines mostly focus on the disassembly process of consumer electronics. Reference [10] developed a disassembly job shop capable of handling both tiny consumer electronics (e.g., telephones, computers, microwave ovens, and hair dryers) and bigger consumer electronics (e.g., refrigerators, television sets, and computer monitors). The suggested configuration comprises two paced disassembly lines and one unpaced parallel disassembly line. According to the diagram, both lines are positioned on opposite sides of a common outgoing conveyor belt that transports the dismantled parts and components to a final sorting area. When designing a plan, it is crucial for the designer to consider the arrangement of places that are essential for the disassembly lines. The suggested design employed an elevated hoist and rail system to reduce the need for manual lifting of bulky technological objects. Conversely, the belt conveyor was employed for the transportation of small electronic articles.

Reference [11] developed a mixed integer linear programming model to address a mixed-model two-sided disassembly line balance problem. The model is suitable for products of medium and large sizes, such as vehicles, refrigerators, and air conditioners. An AND/OR graph was employed to illustrate the configuration of discontinued items. The memetic algorithm was adapted to achieve near-optimal solutions for large-scale issues. The proposed method was evaluated against a genetic algorithm and the Gurobi solver. The study revealed that the proposed model is a viable alternative to the genetic algorithm when dealing with medium and large-scale problems.

Reference [12] sought mixed-model two-sided disassembly using a new evolutionary approach, a genetic flatworm algorithm. The problem was formulated as a stochastic model where uncertainty disassembly time was accounted for. The model aims to minimize opened workstation cost, weighted smoothness index, and total hazard index. Two sample products with different stochastic task times were used to test the proposed solver compared with a conventional flatworm algorithm and a modified discrete flower pollination algorithm. The study revealed that the proposed algorithm outperformed the competing algorithms, resulting in superior answers. Additionally, it produced a more outstanding grade of Pareto optimal solutions.

Reference [13] presented an alternative perspective on the mixed-model disassembly line balance problem. Several robots were assigned to workstations. Hence, their qualities and adaptability were taken into account. The main contribution of this paper is not a complex solving approach but rather an effective problem modeling, which is crucial. A comparative analysis was conducted between the problem-specific bi-criterion evolutionary algorithm (PBEA), which incorporates both a non-Pareto criterion and a Pareto criterion, and three other algorithms: NSGA-II, IBEA, and MOEA/D. The hypervolume, a widely used statistic in evolutionary multi-objective optimization problems, was employed to quantify the performance of the algorithms.

Reference [14] investigated the idea of semi-destructive disassembly using mixed-model disassembly lines. A dual-sided disassembly line was specifically engineered to cater to bulky items such as refrigerators and autos. The uncertainty variables in question were the end-of-life product conditions, specifically corrosion and deformation. Therefore, the decision was made to adopt the partial destructive mode. A multi-objective mathematical model was developed to minimize the number of active work cells, the workload evenness, and the cost. The researchers enhanced the non-dominated sorting genetic algorithm (NSGA-II) to tackle the problem. In contrast to the non-destructive method, the partial destructive method can increase profitability while still achieving a satisfactory smoothness index.

3. Mathematical Formulation

The mathematical model described in this section was extended from [11]. However, the dummy nodes which represent AND/OR type relationship constraints are not necessary since it explicitly occurs in practical situations.

3.1 Notation

nD = number of total dummy nodes, $nD = \max\{nD_m\}$.

nN = total number of regular nodes (task), $nN = \max\{nN_m\}$.

nJ = number of available paired workstations.

nM = quantity of product variations.

k = indice of artificial nodes $k \in \{1,,nD\}$.

i, h = indice of regular nodes (tasks) $i, h \in \{1,,nN\}$.

j = indice of paired workstations $j \in \{1,..,nJ\}$.

s = indice of workstation sides $s \in \{1,2\}$.

m = indice of product variations $m \in \{1,,nM\}$.

$PRE[set]$ = the immediate predecessors of an artificial node k for any model m are the regular nodes i.

$SUC[set]$ = the immediate successors of an artificial node k for any model m are the regular nodes i.

θ = station-sides s where regular nodes for any product variations m.

nD_m = the quantity of artificial nodes for any product variations m.

nN_m = the quantity of regular nodes for any product variations m.

C = takt time.

t_{im} = disassembly time of a regular task i for any product variations m.

F_j = if paired workstation j is opened from both sides, 1; otherwise, 0.

G_j = if paired workstation j is opened from only one side, 1; otherwise, 0.

U_{js} = if paired workstation j is opened from the workstation side s, 1; otherwise, 0.

δ_{hi} = if task h is completed prior to task i, 1; otherwise, 0.

z_{mi} = if task i is of product variation m is selected as the first rank in AND/OR graph, 1; otherwise, 0.

γ_{mjs} = if workstation side s of paired workstation j for product variation m is opened, 1; otherwise, 0.

x_{mijs} = if task i for product variation m is allocated to the workstation side s of paired workstation j, 1; otherwise, 0.

3.2 Mathematical model

Objective function

$$\min\ z_1 = \sum_{i=1}^{nJ} (j \times (F_j + G_j)), \tag{3.1}$$

$$\min\ z_2 = \epsilon \times \sum_{j=1}^{nJ} \sum_{s=1}^{nS} (U_{js}), \tag{3.2}$$

$$\min\ z_3 = \sqrt{\frac{\sum_{j\in J} \sum_{s=1,2} (\max_{j\in J, s=1,2} \{T_{js} - T_{js})^2}{\sum_{j\in J} \sum_{s=1,2} U_{js}}}, \tag{3.3}$$

Subject to

$$\sum_{i\in SUC(m,k)} z_{mi} = 1,\ \forall m, \forall k \in \{0\}, \tag{3.4}$$

$$\sum_{i\in SUC(m,k)} z_{mi} = \sum_{i\in PRE(m,k)},$$
$$\forall m, \forall k \in \{0\}, \tag{3.5}$$

$$\sum_{j=1}^{nJ} \sum_{s\in\theta(i)} x_{mijs} = z_{mi}, \forall m, \forall k \in \{1, , nN_m\} \tag{3.6}$$

$$\sum_{i\in PRE(m,k)} \sum_{v=1}^{j} \sum_{(s\in\theta(i)} x_{mivs}$$
$$\geq \sum_{i\in SUC(m,k)} \sum_{s\in\theta(i)} x_{mijs},$$
$$\forall m, \forall k \in \{1, , nD_m\}, \tag{3.7}$$

$$tf_{mi} \leq C \times z_{mi}, \forall m, \forall i \in \{1, , nN_m\}, \tag{3.8}$$

$$tf_{mi} \geq t_{mi} \times z_{mi}, \forall m, \forall i \in \{1, , nN_m\}, \tag{3.9}$$

$$f_{mi} - tf_{mh} + M \times (2 - \sum_{s\in\theta(i)} x_{mijs} -$$
$$\sum_{s\in\theta(h)} x_{mhjs}) \geq t_{mi}, \forall m, \forall j, \forall k \in \{1, , nA_n\},$$
$$\forall i \in SUC(m,k), \forall h \in PRE(m,k), \tag{3.10}$$

$$tf_{mi} - tf_{mh} + M \times (3 - x_{mijs} - x_{mhjs} - \delta_{hi})$$
$$\geq t_{mi}, \forall m, \forall j, \forall s, \forall i, h \in \{i > h | 1, , nN_m\}, \tag{3.11}$$

$$tf_{mh} - tf_{mi} + M \times (2 - x_{mijs} - x_{mhjs} - \delta_{hi})$$
$$\geq t_{mh}, \forall m, \forall j, \forall s, \forall i, h \in \{i > h | 1, , nN_m\}, \tag{3.12}$$

$$\sum_{i\in\theta(s)\cap i\leq nN_m} x_{mijs} - nN_m \times \gamma_{mjs} \leq 0,$$
$$\forall m, \forall j, \forall s, \tag{3.13}$$

$$\sum_{m=1}^{nM} \gamma_{mjs} - nM \times U_{js} \leq 0, \forall j, \forall s, \tag{3.14}$$

$$\sum_{m=1}^{nS} U_{js} - 2 \times F_j - G_j = 0, \forall j, \tag{3.15}$$

$$F_j, G_j, U_{js}, z_{mi}, x_{mijs}, \delta_{hi}, \gamma_{mjs} \in \{0, 1\},$$
$$\forall m, \forall i, h, \forall j, \forall s, \tag{3.16}$$

$$tf_{mi} \geq 0, \forall m, \forall i. \tag{3.17}$$

The initial term of the objective Eq. (3.1) models the configuration of a compact disassembly production line. Eq. (3.2) aims to minimize the overall number of activated stations while prioritizing jobs with the shortest processing time. Eq. (3.3) represents a measure of workload smoothness, aiming to minimize the disparity in operating time across workstations. This approach promotes fairness in job distribution and effectively enhances production efficiency. Eq. (3.4) guarantees that only one of the successors ($i \in SUC(m, 0)$) of the artificial node, which holds the highest rank in the precedence graph of each m model, is chosen. Eq. (3.5) is selected from the antecedents of the counterfeit node in the precedence graph of product variation m, where ($i \in SUC(m, k)$) allows for only one choice. Eq. (3.6) guarantees that all regular nodes selected for product variations m are consistently assigned to during one of the available job assignment slots. Eq. (3.7) represents a constraint that defines the order of precedence between two elements. Eq. (3.8) guarantees that the completion times of all chosen regular nodes for product variations m do not surpass the cycle time. Eq. (3.9) guarantees that the completion time of all jobs selected for product variations m is more than or equal to the job duration. Eq. (3.10) guarantees that for product variations m, the time difference between the completion of selected tasks given to the same station and location and having a priority relationship is more than the duration of the preceding job. Eqs. (3.11)-(3.12) guarantee that, for product variations m, the discrepancy in finish time between the selected jobs assigned to the same station and location is equivalent to the processing time. The goal function in Eqs. (3.13)-(3.15) allows choice variables to have values. Eqs. (3.16)-(3.17) specify the nature of the choice variables.

This mathematical model comes up with some assumptions: 1) Task times are deterministic and constant for all single product variation, 2) the disassembly line can process a soft-product variety, 3) once the first task of a product starts, the product must be finished to the end of the process line, and 4) a transformed AND/OR graph is accessible for each product variation inside the process.

4. Proposed Metaheuristic Method

4.1 Particle swarm optimization

Particle swarm optimization (PSO) is a method within the topic of Swarm Intelligence, specifically falling within the domain of Evolutionary Computation (EC). It was initially introduced by [15], and then expanded upon by Kennedy and Eberhart in the same year. The Particle Swarm Optimization (PSO) procedure is an iterative procedure that involves a collection of small things, known as particles, which are iteratively employed to explore the search space of certain functions. The particles evaluate their fitness ratings based on the search function at their current places. Following this, every individual particle determines its trajectory within the exploration area by integrating data regarding its present level of fitness, its optimal fitness achieved in previous positions (from an individual standpoint), and the optimal fitness positions in relation to one or more members of the collective (from a social standpoint), while also incorporating certain random disturbances. The subsequent iteration commences subsequent to the updating of the positions of all particles.

The core principle of the conventional Particle Swarm Optimization (PSO) algorithm is the intelligent exchange of information regarding the best local and global values. The process of updating velocity is contingent upon the acquisition of information from preceding stages of the algorithm. Regarding memory, every particle has the capability to retain the optimal place it has encountered throughout its search procedure. The set P denotes the memory set of the swarm S, where $P = p_1, p_2, ..., p_N$, represents the collection of the best positions of each particle (referred to as local best). The position that is seen as the most optimal by all particles is commonly referred to as the global best. Hence, it is justifiable to store and disseminate this vital information. The $gbest$ algorithm integrates the optimal position variable with the optimal function value in the population P at a specific iteration t.

$$pbest_i = [p_i1, p_i2, , p_iD]^T \in A, i = 1, 2, .., N, \tag{4.1}$$

$$gbest^t = arg \min_i f(pbest_i^t). \tag{4.2}$$

The conventional PSO technique, first developed by [15], can be mathematically represented as follows:

$$v_{ij}^{t+1} = v_{ij}^t + \phi_1\beta_1(pbest_{ij}^t - x_{ij}^t)$$

$$+\phi_2\beta_2(gbest_j^t - x_{ij}^t), \quad (4.3)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1}, \quad (4.4)$$

where $i = 1, 2, , N$ and $j = 1, 2, , D; t$ represents the iteration counter; β_1 and β_2 are random variables uniformly distributed within [0,1]. x_{ij}^t and v_{ij}^t are the position and velocity of particle i in dimension j at iteration t. An extended PSO includes δ in the first term of (3.20) to express an inertial force of the particles.

4.2 Modified PSO for Mixed-model DLBP

There are two points of the traditional PSO that were modified. Firstly, to increase the algorithm's performance, a discrete technique proposed by [16] is applied. This technique was proved to improve search exploration and flee from local optimal solutions. Furthermore, this discretization technique dominates other conventional techniques such as Sigmoid function and Hyperbolic tangent function.

Secondly, since the combinatorial optimization problem rarely tends to find a globally optimal solution, we deploy the survival sub-swarm adaptive particle swarm optimization with velocity-line bouncing (SSS-APSO-vb) [17]. It was proved that in high complex and numerous local optimal search space, the SSS-APSO-vb dominates other fancy search algorithms. The proposed method's procedure is portrayed in Fig. 1.

Please note that the particles' velocity and position updating are based on [16] and the mechanisms of self-adaptation, velocity-line bouncing, and extinction and offspring reproduction are based on [17]. The fitness function evaluation is followed Eqs. (3.1)-(3.3).

4.3 Encoding and decoding schemes

Encoding and decoding are essential processes for employing a metaheuristic methodology. Fig. 2 illustrates the swarm's string, comprising two particles; the string S_1 represents a possible solution. Decoding pertains to the formulation of the disassembly process. The tasks are allocated sequentially according to the string. The location, whether left or right, is contingent upon the facts L and R about the task; see Fig. 3. The task with attribute E is randomly designated to either the left or right side.

This study used the penalty technique to satisfy all constraints. In this technique a penalty is imposed on the objective functions to penalize individuals for constraint violations. We selected a substantial number of M for penalization.

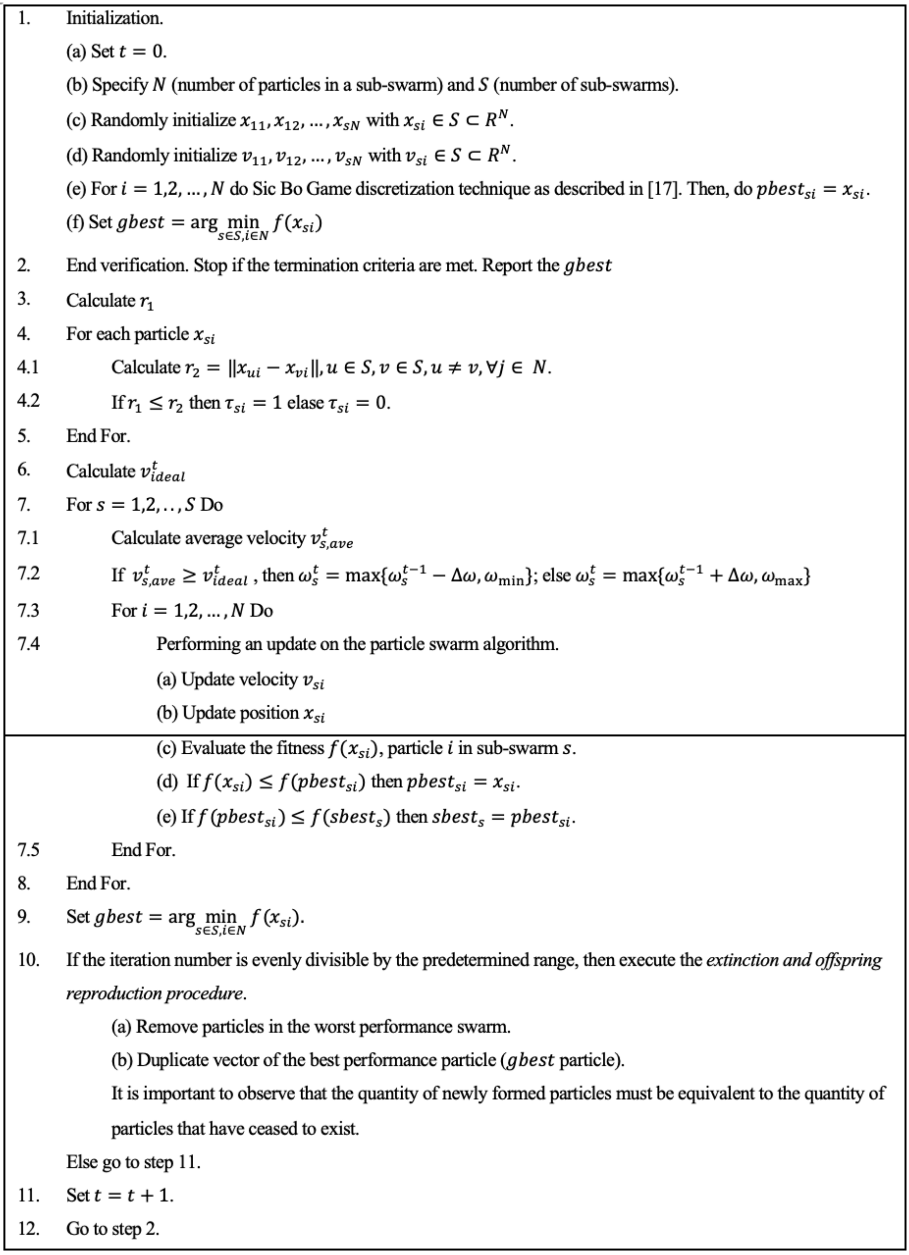
5. Case Experiments

5.1 Top-loaded washing machine

Washing machines fall into the medium-large equipment category and accounted for 24% of global e-waste (53.6 million metric tons) generated in 2020 [18]. Accordingly, an effort to minimize landfill and incinerators has a huge impact on our society. Fig. 4 shows the portion of e-waste generated worldwide in 2020.

Additionally, the medium-large equipment tends to consume a huge volume of landfill and is a cause of high reverse-logistic costs. An efficient disassembly line near the source location can reduce environmentally friendly operation costs. Fig. 5 shows the top-loaded washing machine in this study. Figs. 4-6 illustrate the disassembled parts and subassemblies which included recyclable materials and non-recyclable materials. However, most of them are recyclable ones.

Table 1 lists the disassembly tasks for taking out parts and subassemblies. It also



```
1.    Initialization.
      (a) Set $t = 0$.
      (b) Specify $N$ (number of particles in a sub-swarm) and $S$ (number of sub-swarms).
      (c) Randomly initialize $x_{11}, x_{12}, \dots, x_{sN}$ with $x_{si} \in S \subset R^N$.
      (d) Randomly initialize $v_{11}, v_{12}, \dots, v_{sN}$ with $v_{si} \in S \subset R^N$.
      (e) For $i = 1,2,\dots,N$ do Sic Bo Game discretization technique as described in [17]. Then, do $pbest_{si} = x_{si}$.
      (f) Set $gbest = \arg\min_{s\in S, i\in N} f(x_{si})$
2.    End verification. Stop if the termination criteria are met. Report the $gbest$
3.    Calculate $r_1$
4.    For each particle $x_{si}$
4.1       Calculate $r_2 = \|x_{ui} - x_{vi}\|, u \in S, v \in S, u \neq v, \forall j \in N$.
4.2       If $r_1 \leq r_2$ then $\tau_{si} = 1$ elase $\tau_{si} = 0$.
5.    End For.
6.    Calculate $v_{ideal}^t$
7.    For $s = 1,2,..,S$ Do
7.1       Calculate average velocity $v_{s,ave}^t$
7.2       If $v_{s,ave}^t \geq v_{ideal}^t$ , then $\omega_s^t = \max\{\omega_s^{t-1} - \Delta\omega, \omega_{\min}\}$; else $\omega_s^t = \max\{\omega_s^{t-1} + \Delta\omega, \omega_{\max}\}$
7.3       For $i = 1,2,\dots,N$ Do
7.4           Performing an update on the particle swarm algorithm.
              (a) Update velocity $v_{si}$
              (b) Update position $x_{si}$
              (c) Evaluate the fitness $f(x_{si})$, particle $i$ in sub-swarm $s$.
              (d) If $f(x_{si}) \leq f(pbest_{si})$ then $pbest_{si} = x_{si}$.
              (e) If $f(pbest_{si}) \leq f(sbest_s)$ then $sbest_s = pbest_{si}$.
7.5       End For.
8.    End For.
9.    Set $gbest = \arg\min_{s\in S, i\in N} f(x_{si})$.
10.   If the iteration number is evenly divisible by the predetermined range, then execute the extinction and offspring reproduction procedure.
          (a) Remove particles in the worst performance swarm.
          (b) Duplicate vector of the best performance particle ($gbest$ particle).
          It is important to observe that the quantity of newly formed particles must be equivalent to the quantity of particles that have ceased to exist.
      Else go to step 11.
11.   Set $t = t + 1$.
12.   Go to step 2.
```

Fig. 1. Modified SSS-APSO-vb.

String $S_1 = [2_L\ 3_R\ 5_R\ 1_L\ 7_L\ 10_L\ 8_R\ 9_R\ 4_R\ 6_R]$

String $S_2 = [5_R\ 1_L\ 4_R\ 7_L\ 10_L\ 2_L\ 8_R\ 9_R\ 3_R\ 6_R]$

Fig. 2. Two-solution strings for DLBP.

shows the disassembly time of each product model and precedence constraint. There are four models in this product family. Some of them have no some parts; accordingly, the disassembly time is zero, while some of them have different disassembly time in

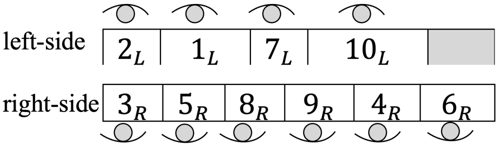


Fig. 3. Decoding schematic of string S_1.

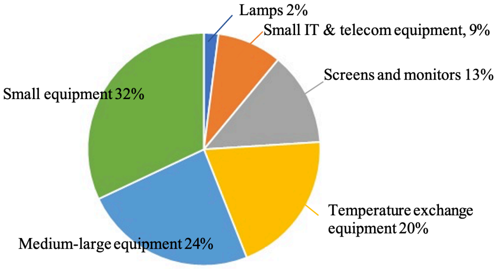


Fig. 4. E-waste generated worldwide in 2020.

Fig. 5. Top-loaded washing machine.

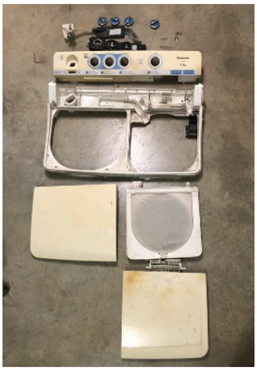


Fig. 6. Disassembly parts I.

the same task. Nevertheless, we need to design a disassembly line that is appropriate for all four models. Please note that tasks 10, 11, and 34 are destructive disassembly operations. The precedence relationship diagram of the case experiment given in Fig. 9 corresponds to the data given it Table 1. Additionally, POR/SOR relationships are also illustrated. Typically, AND/OR relationships have been described in literature. Nevertheless, in this study, POR/SOR which are the sub-types of OR type relations are deployed to reflect the industrial practice. AND type precedence relations between two tasks indicates that one task cannot start before the other finishes. Fig. 11 illustrates the AND type precedence relations.

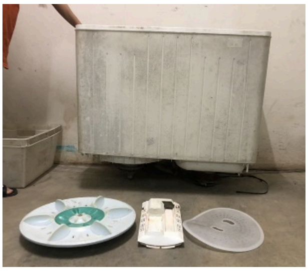


Fig. 7. Disassembly parts II.

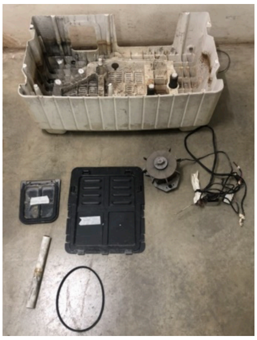


Fig. 8. Disassembly parts III.

Basically, OR type relations are specific to the disassembly network. POR-type relations indicate that no less than one task from a defined set must be completed before the commencement of another work. SOR type relations express that at most one of the tasks in a specific set can be performed after one task completes. Figs. 10 to 12 demon-

Table 1. Top-loaded washing machine.

Task	Part / Direction	Disassembly time by model (sec.)				Precedence part
		A	B	C	D	
1	Bolts (2) / L	5.72	5.72	5.72	5.72	-
2	Bolt (1) / R	6.74	6.74	6.74	6.74	-
3	Cover / E	2.75	2.75	2.75	2.75	-
4	Panel / E	3.36	3.36	3.36	3.36	1, 2
5	Wash timer wiring / L	25.22	25.22	10.52	10.52	4
6	Spin timer wiring / R	38.64	38.64	38.64	45.32	4
7	Washing timer knob / L	3.43	3.43	3.43	3.43	5
8	Spin timer knob / R	3.03	3.03	3.03	3.03	6
9	Timer switch bolts (2) / R	35.61	35.61	35.61	35.61	4
10	Bind Tapping (5) / R (destructive)	32.54	32.54	24.79	20.55	4
11	Bind Tapping (5) / L (destructive)	32.54	32.54	24.79	20.55	4
12	Washing timer switch / L	4.59	4.59	4.59	4.59	10, 11
13	Spin timer switch / R	3.19	3.19	3.19	3.19	9
14	Washing selector knob / L	4.56	4.56	4.56	4.56	13
15	Cycle selector knob / R	3.46	3.46	3.46	0	12, 14
16	Buzzer / R	4.56	4.56	4.56	4.56	10, 11
17	Pannel A / E	4.16	4.16	0	0	12, 13
18	Switch cover / L	3.73	3.73	3.73	3.73	17
19	Spinner lid / R	3.17	3.17	3.17	3.17	17
20	Body b plate bolts (2) / R	10.72	10.72	10.72	10.72	19
21	Nozzle holder bolts (2) / L	3.19	3.19	3.19	3.19	18
22	Body b (3) / L	4.90	4.90	4.90	4.90	3, 19
23	Nozzle holder / L	4.55	4.55	4.55	4.55	21
24	Body b plate / R	3.56	3.56	3.56	3.56	20
25	Body b / R	23.63	23.63	23.63	23.63	22
26	Over flow filter a / L	2.95	2.95	2.95	2.95	25
27	Special bolt / L	17.51	0	0	0	3
28	Pulsator unit / L	2.52	2.52	2.52	2.52	27
29	Back panel bolt (1) / R	2.53	2.53	2.53	2.53	-
30	Back panel / R	10.58	12.36	11.05	11.05	29
31	Back panel bolts (2) / L	9.14	9.14	9.14	9.14	-
32	Back panel / L	2.73	2.73	2.73	2.73	31
33	Base a bolts (3) / R	2.43	2.43	2.43	2.43	-
34	Bolt / R (destructive)	32.07	32.07	32.07	32.07	30
35	Tub a / E	30.77	30.77	30.77	30.77	30, 33
36	Drain tube / L	7.45	7.45	7.45	7.45	30
37	Motor bolts (3) / L	3.7	5.2	5.2	8.9	32, 35
38	Electric wire / L	62.39	70.55	70.55	60.31	30
39	Motor / L	3.69	3.69	7.22	7.22	35, 37

Remark: Parenthesis after the part name is the quantity; R, L, and E are right, left, and equivalence directions, respectively.

strate examples of three types of precedence relations.

According to Fig, 10, tasks i, j, and k are predecessors of task l. These three tasks must be completed before task l starts. In Fig. 11, no less than one task in set $\{a, b, c\}$ must be completed before task d begins. Nevertheless, in Fig. 12, the tasks in set $\{x, y, x\}$ are OR successors of task w. As a result, after completing task w, at most one of the tasks in the set can be performed.

5.2 Parameter settings

A drawback of the metaheuristic approach is that there are many parameters that need to be predetermined [19]. In this study, we will cultivate the previous study by using their initial parameter settings in our numerical experiment. Table 2 shows the details of parameter settings of the proposed method. It is not guaranteed that these settings will yield the best solution; however, it is good enough based on the

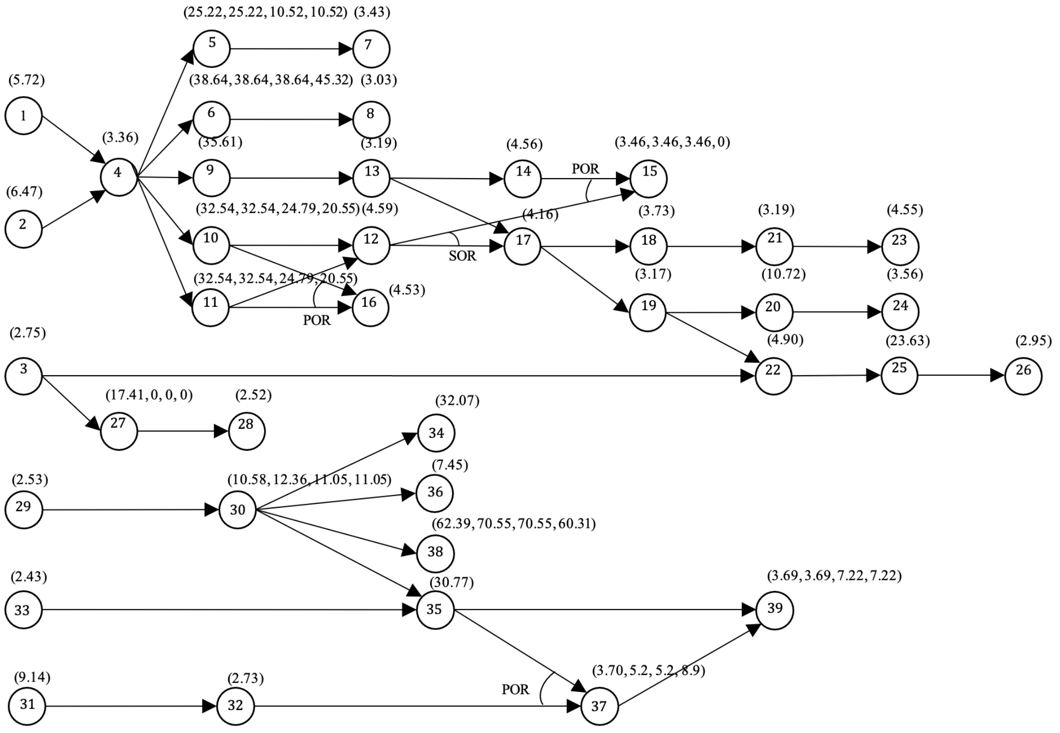


Fig. 9. Top-loaded washing machine disassembly precedence diagram.

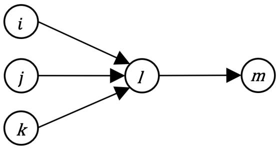


Fig. 10. AND type precedence relations.

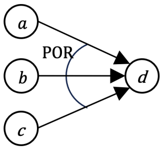


Fig. 11. POR type precedence relations.

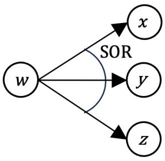


Fig. 12. SOR type precedence relations.

previous study [16].

Table 2. Parameter settings.

Parameter	Description	Value
$w_{\max}$	Maximal inertia weight	0.7
$w_{\min}$	Minimal inertia weight	0.3
Δw	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

5.3 Experiment design

The proposed algorithm was implemented in C++ using Microsoft Visual Studio 2019 and executed on a laptop equipped with an Intel Core i7-8750H CPU @2.20GHz processor and 8 GB RAM, running the Windows 10 operating system.

To evaluate the effectiveness of the

proposed method, we chose two well-regarded algorithms from the existing literature: NSGA-II [20, 21] and SPEA2 [22]. Two solvers on the NEOS server (neos-server.org), BARON and MINLP, were selected to compare with the proposed algorithm. The AMPL language was used to interface with the solvers. Please note that the competitive algorithms are for a multi-objective optimization problem; accordingly, Pareto-optimality theory is used to identify the superior solutions that line the Pareto-optimal front. The algorithms were executed 30 times to eliminate random discrepancy.

A parameter used to test in this study is the takt time. It is a quantitative measure that determines the speed at which a product needs to be finished in order to satisfy client requirements. On the other hand, it is the maximum disassembly time for each workstation in the disassembly line. Since the mathematical model attempts to minimize three objectives; i.e., disassembly line length, total number of opened stations, and workload smoothness; the Pareto principle was used to evaluate the algorithm's performance by counting non-dominated on the Pareto-optimal front in a three-dimensional search space.

The assessment of solution quality is typically done using the inverted generational distance (IGD), hypervolume (HV), and the ratio (R) of the non-inferior solutions of the algorithm (A_i). The IGD and R formulae are shown below.

Let S be a solution set of an algorithm on a given multi-objective optimization problem; W is a collection of evenly distributed representative points to make up the Pareto front. The IGD value of S relative to W can be determined as,

$$IGD = (S, W) = \frac{\sum_{\in C} d(w, S)}{|W|}, \quad (5.1)$$

where $d(w, S)$ is the shortest distance between p and the points in S, and $|W|$ is the cardinality of W. The IGD quantifies the imminence and diversification exhibited by the set of solutions. A lower IGD value indicates a superior approach.

$$HV(S) = vol(\bigcup_{x \in S} [f_1(x), z_1] \times \cdots \times [f_M(x), z_M]), \quad (5.2)$$

where $vol(\cdot)$ is the Lebesgue measure, and $z^p = (z_1, , z_M)^T$ is a given reference point. The hypervolume (HV) quantifies the extent of the objective space controlled by the solutions in set S and limited by z^p. Please note that, in this study, all the HV are normalized to [0,1] by dividing $\Pi_{i=1}^{M} z_i$. The larger HV is the better algorithm.

Let P_i represent the set of Pareto non-inferior solutions generated by algorithm i. Therefore, $P = P_1 \bigcup P_2 \bigcup \cdots \bigcup P_I$ represents the collection of non-inferior solutions generated by all competitive algorithms. The ratio (R) can be computed by dividing the number of non-inferior solutions of each algorithm (P_i) by the number of non-inferior solutions of all competitive approaches (P).

$$R(P_i) = \frac{|P_i - \{X \in P_i | \exists Y \in P : Y \prec X\}|}{|P|}. \quad (5.3)$$

Logically, a higher R-value indicates a larger quantity of non-dominated solutions, which in turn indicates a superior algorithm.

6. Numerical Results

Table 3 shows the Pareto front solutions of the five competitive methods with different takt times for 30 independent executions. The predetermined takt times were 71, 80, 90, and 100 seconds, respectively. Please note that the takt time starts at 71 seconds because the most extended task that

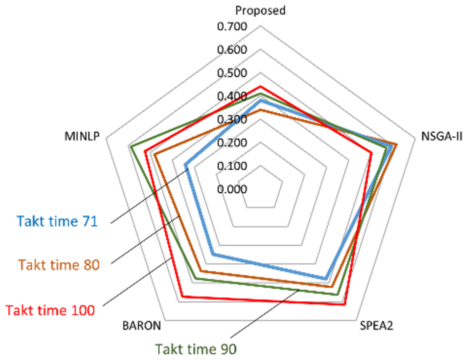


Fig. 13. Coefficient of variation comparison.

cannot be divided is task number 38 of models B and C, which has a task time of 70.55 seconds.

Table 3 shows the non-dominated solutions found so far. Theoretically, many Pareto fronts were found to be superior-performance search approaches. The statistics data comprised the highest, lowest, average, and standard deviation from thirty independent executions. The coefficient of variation ($\hat{C}_v$) is a statistical measure that standardizes the dispersion of a probability distribution. The term refers to the division of the standard deviation by the mean (δ/μ). A lower coefficient of variation indicates that the method is more consistent compared to a higher coefficient of variation.

Fig. 13 is the radar chart of $\hat{C}_v$. As mentioned, the low coefficient of variation means a consistent search algorithm, which is preferable. The proposed method shows superior performance with high consistency for cases of takt time. BARON and SPEA2 have high variation; besides, it shows they are good at exploring. However, they have a low $\hat{C}_v$ at takt times of 71 and 80 sec., while they have a high $\hat{C}_v$ at takt times of 90 and 100 sec.

NSGA-II is different. It has a high $\hat{C}_v$ in all cases. This shows that the NSGA-II is able to explore the search space considerably; nevertheless, it lacks exploitation ability. MINLP yielded no competitive results because it cannot solve non-convex substantially.

Likewise, the proposed approach worked well at the ratio indicator by providing the outstanding ratio in 71, 80, and 100-sec cases. Nevertheless, the NSGA-II worked well with the 90 sec takt time case. Since most of the algorithms in this study are search methods, we need to conduct more statistical investigation.

Table 4 shows the three measurements: IGD, HV, and R. Five competitive algorithms solved the instance in four predetermined takt times; 71, 80, 90, and 100 sec, respectively. Each combination was executed thirty times, and the three measurements were determined. The bold numbers in the table mean the best in its combination.

From Table 4, our proposed algorithm provided outstanding solutions in all cases in the inverted generational distance indicator. This implies that the proposed method is powerfully effective in exploring search space. Additionally, the proposed algorithm dominated other competitive algorithms at the hypervolume indicator; however, the MINLP outperformed our algorithm in the case of 100 sec.

Two-way analysis of variance (Two-way ANOVA) was used to examine some hidden implications from the experiments. The null and alternative hypotheses were H_0: all algorithms are not significantly different with the specific indicator, and H_1: at least one algorithm is significantly different from others.

Table 5 is the ANOVA table for algorithm types, with takt times as the factors and IGD as the response. The table shows that the algorithms were not different from

Table 3. Computational results.

Takt time (sec.)	Pareto front solutions	Modified approach	NSGA-II	SPEA2	BARON	MINLP
71		13	17	18	19	10
		3	3	2	3	2
		8.600	8.567	11.467	13.033	4.700
		3.265	5.070	5.488	4.537	1.601
		0.380	0.592	0.479	0.348	0.341
80		15	17	18	19	11
		4	2	2	3	2
		9.533	8.167	9.633	12.533	4.433
		3.235	5.025	5.034	5.482	2.128
		0.339	0.615	0.523	0.437	0.480
90		16	17	18	19	13
		4	3	2	3	2
		9.767	7.300	10.100	9.733	4.631
		4.006	4.162	5.689	4.631	4.310
		0.410	0.570	0.563	0.476	0.588
100		17	17	18	19	13
		4	2	3	4	3
		10.767	7.600	6.700	10.267	7.600
		4.739	3.811	4.120	5.889	3.979
		0.440	0.501	0.615	0.574	0.524

Table 4. Three-dimensional measurement comparison.

Takt time (sec)	Algorithm	IGD	HV	R
71	Proposed	**5.33E+01**	**9.74E-01**	**3.57E-02**
	NSGA-II	2.13E+02	8.36E-01	2.15E-02
	SPEA2	1.09E+02	7.98E-01	1.76E-02
	BARON	8.39E+01	8.02E-01	6.30E-03
	MINLP	1.04E+02	8.95E-01	9.10E-03
80	Proposed	**6.22E+00**	**9.32E-01**	**4.23E-02**
	NSGA-II	7.36E+00	8.80E-01	3.32E-02
	SPEA2	8.44E+00	8.90E-01	3.78E-02
	BARON	7.74E+00	8.92E-01	1.75E-02
	MINLP	8.05E+00	9.02E-01	9.30E-03
90	Proposed	**4.39E+00**	**9.58E-01**	1.68E-01
	NSGA-II	5.32E+00	9.53E-01	**1.91E-01**
	SPEA2	5.07E+00	9.03E-01	1.71E-01
	BARON	7.03E+00	8.83E-01	1.08E-01
	MINLP	8.30E+00	9.88E-01	1.00E-01
100	Proposed	**4.06E+00**	9.69E-01	**1.83E-01**
	NSGA-II	5.00E+00	9.08E-01	1.28E-01
	SPEA2	4.06E+00	9.03E-01	1.52E-01
	BARON	6.90E+00	8.85E-01	1.18E-01
	MINLP	7.05E+00	**9.94E-01**	1.08E-01

Remark: the bold numbers are the best result among competitive algorithms.

the inverted generational distance indicator (at $\alpha = 0.05$).

Tables 6-7 are the ANOVA tables on hypervolume and ratio indicators, respectively.

Table 5. Two-way ANOVA on IGD.

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Algorithm	4	3670	917.5	1.01	0.443
Takt time	3	42535	14178.2	15.53	0.000
Error	12	10955	912.9		
Total	19	57159			

Table 6. Two-way ANOVA on HV.

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Algorithm	4	0.02823	0.007057	7.38	0.003
Takt time	3	0.01842	0.006140	6.42	0.008
Error	12	0.01147	0.000956		
Total	19	0.05812			

Table 7. Two-way ANOVA on R.

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Algorithm	4	0.007748	0.001937	6.19	0.006
Takt time	3	0.072051	0.024017	76.76	0.000
Error	12	0.003754	0.000313		
Total	19	0.083553			

Table 6 shows the p-value = 0.003, while the significant level, α, is 0.05. Thus, at least one algorithm differs from others regarding the hypervolume indicator. Moreover, Table 7 indicates a p-value = 0.006, while the significant level, α, is 0.05. This means that at least one algorithm differs from others in terms of the ratio indicator.

Subsequently, we performed Tukey's test to compare the means of each treatment with the means of all other treatments. The comparison results for the hypervolume and ratio indicators are shown in Tables 8-9, respectively, with the confidence level set at 95%.

Table 8. Grouping on IGD using Tukey test.

Algorithm	N	Mean	Grouping	
Proposed	4	0.958200	A	
MINLP	4	0.944400	A	
NSGA-II	4	0.894125	A	B
SPEA2	4	0.873225		B
BARON	4	0.865400		B

Remark: Means that do not share a letter are significantly different.

Table 9. Grouping on HV using Tukey test.

Algorithm	N	Mean	Grouping	
Proposed	4	0.107150	A	
MINLP	4	0.094500	A	B
NSGA-II	4	0.093400	A	B
SPEA2	4	0.062625		B
BARON	4	0.056575		B

Remark: Means that do not share a letter are significantly different.

Table 9 demonstrates that our proposed method, SPEA2 and NSGA-II, are in the same superior group. Nevertheless, SPEA2 and NSGA-II are similar to the lower group since they can fit both groups.

In conclusion, based on three indicators, the proposed algorithm worked well in all cases. It outperformed all indicators and consistently yielded the solution. It may be argued that the competitive algorithms also worked well in some cases. Nonetheless, the proposed algorithm showed robustness by persistently yielding promised solutions.

7. Small-to-Medium Case Examples

To illustrate the proposed technique's performance, an electric rice cooker with two models in the family and a microwave oven with three models were experimented with (see Figs. 16-17). Figs. 18 and 19 show the assembly tasks of the electric rice cooker and the microwave oven, respectively. There are twenty-six disassembly tasks for the electric rice cooker and thirty-six for the microwave oven. Please note that all the direction attributes for the electric rice cooker are E while ten tasks of microwave oven disassembly are E.

Fig. 14. Electric rice cooker.

Fig. 15. Microwave oven.

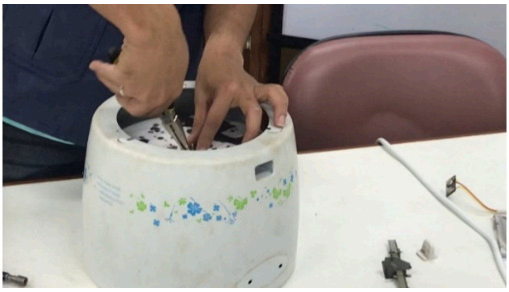


Fig. 16. Disassembly tasks of the electric rice cooker.

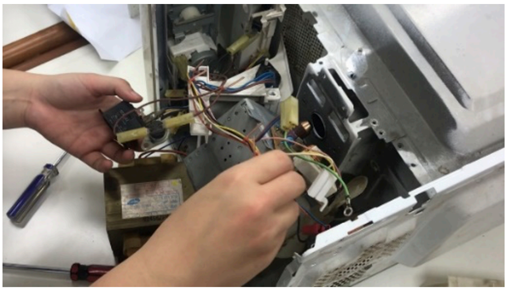


Fig. 17. Disassembly tasks of the microwave oven.

Tables 10-11 illustrate the three-dimensional measurement results of the electric rice cooker and microwave oven, respectively.

Table 10. Three-dimensional measurement comparison for electric rice cooker.

Takt time (sec)	Algorithm	*IGD*	*HV*	*R*
60	Proposed	**3.15E+01**	4.14E-01	3.21E-02
	NSGA-II	7.84E+01	1.45E-01	2.70E-02
	SPEA2	2.15E+02	1.33E-01	4.39E-03
	BARON	1.27E+02	3.77E-01	7.02E-03
	MINLP	3.39E+02	**4.53E-01**	**3.55E-02**

Remark: the bold number are the best result among competitive algorithm.

Table 11. Three-dimensional measurement comparison for microwave oven.

Takt time (sec)	Algorithm	*IGD*	*HV*	*R*
120	Proposed	**4.17E+01**	7.07E-01	4.30E-02
	NSGA-II	7.32E+01	4.12E-01	3.64E-02
	SPEA2	7.95E+01	5.08E-01	2.55E-02
	BARON	6.33E+01	6.79E-01	9.72E-03
	MINLP	1.04E+02	**5.89E-01**	**7.81E-03**

Remark: the bold number are the best result among competitive algorithm.

Table 10 shows that the MINLP is a highly competitive search technique. The proposed method could not yield a significantly different result when the disassembled product is small, especially if its structure is not complicated. The medium product has some intricate patterns, as the microwave oven is highly recommended for using our proposed method. Table 11 shows the outstanding measurements.

8. Conclusion

The mathematical model was proposed. The model minimizes the disassembly line length, number of opened stations, and workload smoothness. It was invented to solve the mix-model disassembly line balancing problem; furthermore, it could solve two-sided disassembly line balancing. The problem fell into a multi-objective optimization problem. Thus, a modified particle swam optimization solver was invented to search the Pareto front as many times as possible.

The top-loaded washing machines with four models in the series were used as the benchmark instances with four predetermined takt times. These were compared with four competitive algorithms in the literature, invented to solve combinatorial and multi-objective optimization problems. The computational results demonstrated that the suggested method surpassed the competitive solvers in this study by providing promising solutions in all three indicators: the inverted generational distance, hypervolume, and ratio. Furthermore, it exhibited robust performance by providing a low coefficient of variation values. Additionally, the proposed method yielded reasonable solutions for medium disassembly products while showing the acceptability for small disassembled products.

Nevertheless, this study should have compared other characteristics of meta-heuristics methods, such as convergence rate, computational time, and explicit exploration and exploitation performances. Thus, in future research, the researcher is interested in investigating issues by preparing more reliable computational machines. Additionally, the researcher intends to use the disassembly line design with hazardous materials incorporated with partial destructive processes.

Acknowledgements

This research project was granted by the Silpakorn University Research, Innovation, and Creative Fund. We would like to thank the anonymous reviewers for their thoughtful and constructive feedback. This article was significantly improved by their invaluable comments.

References

[1] Özceylan E, Kalayci CB, Güngör A, Gupta SM. Disassembly line balancing problem: a review of the state of the art and future directions. Int J Prod Res. 2019;57(15-16):4805–27.

[2] Güngör A, Gupta SM. Disassembly line in product recovery. Int J Prod Res. 2002;40(11):2569–89.

[3] Gupta SM, Taleb KN. Scheduling disassembly. Int J Prod Res. 1994;32(8):1857–66.

[4] Ghandi S, Masehian E. Review and taxonomies of assembly and disassembly path planning problems and approaches. Comput Aided Des. 2015;67:58–86.

[5] Güngör A, Gupta SM, Pochampally K, Kamarthi SV. Complications in disassembly line balancing. In: Environmentally Conscious Manufacturing. Vol. 4193. SPIE; 2001. p. 289–98.

[6] Altekin FT, Akkan C. Task-failure-driven rebalancing of disassembly lines. Int J Prod Res. 2012;50(18):4955–76.

[7] McGovern SM, Gupta SM. Disassembly Line Balancing. In: Environment Conscious Manufacturing. CRC Press; 2007. p. 235–310.

[8] Hu Y, Xu Y, Jia Y. Review of the optimization algorithms for remanufacturing disassembly line. In: Proc 26th Int Conf Automation and Computing (ICAC). IEEE; 2021. p. 1–6.

[9] McGovern SM, Gupta SM. Disassembly Line: Balancing and Modelling. New York: McGraw-Hill; 2011.

[10] Opalić M, Kljajin M, Vuĉković K. Disassembly layout in WEEE recycling process. Strojarstvo. 2010;52(1):51–8.

[11] Mutlu S, Güner B. A memetic algorithm for mixed-model two-sided disassembly line balancing problem. Procedia CIRP. 2021;98:67–72.

[12] Liang J, Guo S, Xu W. Balancing stochastic mixed-model two-sided disassembly line using multiobjective genetic flatworm algorithm. IEEE Access. 2021;9:138067–81.

[13] Laili Y, Wang Y, Fang Y, Pham DT. Solutions for mixed-model disassembly line balancing with multi-robot workstations. In: Optimisation of Robotic Disassembly for Remanufacturing. 2022. p. 153–80.

[14] Chao B, Liang P, Zhang C, Guo H. Multiobjective optimization for mixed-model two-sided disassembly line balancing problem considering partial destructive mode. Mathematics. 2023;11(6):1299.

[15] Kennedy J, Eberhart R. Particle swarm optimization. In: Proc ICNN'95 – Int Conf Neural Networks. Vol. 4. IEEE; 1995. p. 1942–8.

[16] Pornsing C, Sangkhiew N, Sakonwittayanon P, Jomtong P, Ohmori S. A new discretization technique for enhancing discrete particle swarm optimization's performance. Sci Technol Asia. 2022;27(3):204–15.

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

[18] Forti V, Baldé CP, Kuehr R, Bel G. The Global E-Waste Monitor 2020. Bonn/Geneva/Rotterdam: United Nations University (UNU), International

Telecommunication Union (ITU) & International Solid Waste Association (ISWA); 2020.

[19] Tongchan T, Pomsing C, Tonglim T. Harmony search algorithm's parameter tuning for traveling salesman problem. In: Proc Int Conf Robotics and Automation Sciences (ICRAS). IEEE; 2017. p. 199–203.

[20] Zuo X, Chen C, Tan W, Zhou M. Vehicle scheduling of an urban bus line via an improved multiobjective genetic algorithm. IEEE Trans Intell Transp Syst. 2014;16(2):1030–41.

[21] Deb K, Pratap A, Agarwal S, Meyarivan TA. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput. 2002;6(2):182–97.

[22] Zhou A, Zhang Q, Zhang G. A multiobjective evolutionary algorithm based on decomposition and probability model. In: Proc IEEE Congress on Evolutionary Computation. IEEE; 2012. p. 1–8.