IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 12, Issue 6, June. 2022, || Series -I || PP 10-21

Fish School Search Optimization Algorithm for Solving U-Shaped Sequence-Dependent Disassembly Line Balancing Problem with Multiple Objectives

Pengfei Yao[1], Surendra M. Gupta[1, *]
[1]*(Department of Mechanical and Industrial Engineering, Northeastern University, USA)*
Received 26 May 2022; Accepted 08 June 2022

Abstract:
Global supply chains have developed rapidly due to the desire for increased economic improvement and manufacturing productivity. However, increased prosperity and the need for latest technology have shortened the lifespan of products, especially electronic products. An increasing number of end-of-life (EOL) products is becoming one of the main problems for pollution and waste. In order to achieve the goal of sustainable development, ecofriendly methods should be encouraged. As opposed to the traditional landfilling method of disposing of EOL products, product recovery is way more environmentally friendly. Remanufacture, reuse, and recycle are popular options of product recovery. For most of the product recovery options, disassembly is often the first step. Large scale disassembly of EOL products is performed on a paced disassembly line. Since U-shaped disassembly line contains many more challenges compared to the traditional straight-line configuration, there is a limited amount of research available about U-shaped layout. Balancing the disassembly line is one of the most crucial problems in the disassembly research area. Due to the existing situation of mutual interaction among tasks, this paper proposes a mixed-integer non-linear programming (MINLP) model to solve the sequence-dependent DLBP on a U-shaped layout (SUDLBP) with the consideration of four different objectives. Because of the NP-hard nature of DLBP, a novel meta-heuristic algorithm, fish school search (FSS) algorithm, is proposed to help find near-optimal solutions. Two instance sets, which contain small-size and large-size benchmark problems, are used to test the performance of the FSS algorithm. Results illustrate that FSS algorithm performs better compared to other algorithms reported in the literature and thus can serve as a great alternative in DLBP field. The results also show that the U-shaped line improves line smoothness and efficiency compared to the straight-line layout.
Key Word: Waste management, Remanufacturing, Disassembly Line Balancing, U-Shaped Disassembly line, Invasive Weed optimization (IWO).

I. INTRODUCTION

Waste and pollution problem is accelerating in this world since product prosperity greatly shorten lifespan of a product and unreasonable treatment of end-of-life (EOL) products[1,2,3]. In the meanwhile, EOL product especially electronic and electricity products contain hazardous part(s) and reusable or recyclable materials, therefore, methods dealing with EOL products should consider environmental protection and cost-based objectives. The consciousness of sustainable development is gradually accepted in this world and environmental protection methods are gaining rapid development. Product recovery is a complete system process to dispose of waste products and it aims to minimize resource waste and maximize environmental protection. Gungor and Gupta (1999b)[4] first highlighted the concept of environmentally conscious manufacturing and product recovery (ECMPRO) with the goal of adding green concept to the whole lifecycle of a product. Remanufacturing, recycle, and reuse are three different methods in product recovery. However, one important step of these methods is disassembly. Disassembly is the first step dealing with EOL products and it aims to complete the separation of a product into its different parts[5]. A disassembly line distributes tasks successively to linked workstations along the conveyor and operators and/or intelligent robotics work inside the workstation. All EOL product will be operated on a disassembly line, therefore, in order to achieve objectives, balancing the disassembly line is a key issue. With rapid development of technologies and the consideration of COVID-19 pandemic, robot-control systems, artificial intelligence, and automatic operation are becoming active research and real-world industrial topic[6].

Disassembly line balancing problem (DLBP) was for the first time proposed by Gungor and Gupta (1999a)[7]. DLBP aims to optimally assign tasks to workstations within the domain of cycle time and precedence relationship constraints. Multiple criteria should be taking into account, viz., types of EOL products, line type, uncertainty of task processing time, objective functions, mathematical model, constraints, algorithms/approaches, complications, and interactions with real world industrial situation[8]. After the pioneering work of Gungor and Gupta, DLBP has become an active research area and attracts more attentions[9]. Sequence-dependent DLBP (SDLBP) is a special type of DLBP with the considering of mutual interference among tasks[10]. Different task processing times will cause different task sequences in the process of optimization and ultimately affect the disassembly line balancing. Typically, there are four types of a disassembly line, viz., straight-line, U-shaped, parallel, and two-sided. The most distinguish characteristic of a U-shaped line is operators and/or robotics can work across the workstation and each workstation has two sides which are entrance side and exit side. This configuration is more complicated that a straight-line layout and may create more optimal task assignments[11].

Exact methods and optimization approaches are gradually proposed on a disassembly line and after the milestone research of McGovern and Gupta (2007a, 2007b)[12,13] which proved that DLBP belongs to NP-hard class problem, heuristics and meta-heuristics are continually applied on a disassembly line[14]. Kalayci and Gupta[10,15,16,17,18] introduces five optimization algorithms viz., artificial bee colony (ACO), particle swarm optimization (PSO), ant colony optimization (ACO), simulated annealing algorithm (SA), tabu search algorithm (TS) and has for the first time studied sequence-dependent DLBP with multiple objectives. Recently, Yao and Gupta (2021a, 2021b, 2021c, 2021d, 2021e, 2021f)[5,9,11,19,20,21] proposed five novel meta-heuristic algorithms viz., cat swarm optimization (CSO), small world optimization (SWO), ant colony optimization (ACO), invasive weed optimization (IWO), teaching-learning-based optimization (TLBO), and fish school search optimization (FSS) on a U-shaped disassembly line which expanded the approach filed of U-shaped DLBP.

The remainder of the paper is organized as follows. Literature review is described in Section 2. Section 3 contains problem definition and the proposed MINLP model with four different objectives. Section 4 provides detailed encoding and decoding procedures of SUDLBP and steps of FSS algorithm. Section 5 presents detailed case study and comparative study. Finally, conclusions and future research directions are provided in Section 6.

II. LITERATURE REVIEW

A U-shaped disassembly line allows operators work across the workstation which may offer more task assignments. Agrawal and Tiwari (2008)[22] first studied U-shape DLBP, and they introduced a collaborative ant colony algorithm to deal with stochastic task times. Avikal and Mishra (2012)[23] proposed a heuristic approach on a U-shaped disassembly line. Avikal, Jain, and Mishra (2013)[24] introduced another U-shaped heuristic on a disassembly line which expand the approach field of DLBP. Partial U-shaped DLBP was first considered in research Wang, Gao, and Li (2020)[25], and later Li and Janardhanan (2021)[26], Wang et al. (2021)[27], Wu et al. (2021)[28] and Lu et al. (2021)[29] proposed novel approaches on a U-shaped layout with the considering of partial disassembly mode. Another special disassembly, sequence-dependent U-shaped DLBP has for the first time considered in research Li, Kuckkkoc, and Zhang (2019)[30] which greatly promote the development of SUDLBP. As mentioned in previous section, Yao and Gupta (2021a, 2021b, 2021c, 2021d, 2021e, 2021f) first proposed 5 novel meta-heuristic algorithms which offers more approach selections in DLBP field.

Sequence-dependent situation considers uncertainty of task processing time since interactions may exist among tasks (Kalayci and Gupta, 2013a; Li et al., 2021[31]). Task i and task j have sequence dependency means additional task processing times should be added based on the order of these involved tasks. This situation often happens between tasks which have no direct precedence relationship (Kalayci and Gupta, 2013b; Wang et al., 2021[32]). If task i and task j have sequence dependency and task processing times for task I and j are t_i, andt_j respectively, the calculation of sequence-dependent task processing time is $t_i + t_j + sd_{ji}$ if task i is disassembled before task j or $t_j + t_i + sd_{ij}$ if task j is removed before task i.

To fill the gap of limited studies of U-shaped DLBP and sequence-dependent UDLBP, this paper considers multi-objective SUDLBP and develops a novel meta-heuristic algorithm. The main contributions of this paper are presented as follows.

(1) Since one objective is non-linear, a mixed-integer nonlinear programming (MINLP) mathematical model is formulated to help solve multi-objective U-shaped DLBP. Related constrains of this model are capable of solving AND precedence and OR precedence relationships.

(2) Fish school search optimization algorithm (FSS), has for the first time, applied to help find near-optimal solutions of sequence-dependent U-shaped DLBP (SUDLBP). A strong ability of balancing between exploration and exploitation makes FSS algorithm suitable for solving multi-objective optimization problem.

(3) A comprehensive comparative study is conducted on two sets of instances in this paper to evaluate the performance of developed MINLP model and the proposed FSS algorithm. Total of 47 benchmark problems are included in instance sets. Results of case studies indicate that U-shaped disassembly line has greater performance on lane smoothness aspect and number of workstations compared with traditional straight-line disassembly line. The comparative study demonstrates that the proposed FSS algorithm outperforms other meta-heuristic algorithms on many aspects.

III. PROBLEM DEFINITION

Sequence-dependent U-shaped DLBP has two complicated challenges than normal straight-line DLBP. First, sequence dependent situation combines the consideration of uncertainty. Task processing time is not deterministic all the time, in sequence-dependent environment, two tasks may create interaction, that means sequence dependency should be added to one of them, therefore, the actual task processing time may change based on instance information. Second, a U-shaped disassembly line has entrance and exit sides, which increase the change of finding more near-optimal task assignments especially dealing with large-size problem. In the proposed model, each workstation contains two sub-stations to classify the side of assigned tasks. To optimally obtain near-optimal solutions, assumptions should be noticed. EOL products are similar and only one type. All parts of the product should be disassembled.

3.1 Model formulation

This section presents MINLP model for SUDLBP with the consideration of AND/OR precedence. Indices and parameters are shown as follows.

Notations	
i,j	Task index, $i, j = 1,2, \ldots, N$
M	Number of workstations
m	Workstation (sub-station) index, $m = 1,2, \ldots, 2M$
t_i	Processing/removal time of task i
h_i	Binary variable, 1, if task i is hazardous; 0, otherwise
d_j	Demand value of task j
ANDP(i)	Set of AND predecessor of task i
ORP(i)	Set of OR predecessor of task i
CT	Cycle time
T_m	Total task processing times of workstation m
sd_{ij}	Sequence dependent time between task j and task i
F_a	Objective function, $a = 1,2,3,4$
Decision variables	
x_{im}	Binary variable, 1, if task i is assigned to sub-station m; 0, otherwise
x'_{imj}	Binary variable, 1, if task is assigned to sub-station m and is operated before task j; 0, otherwise
w_{ij}	Binary variable, 1, if task i is operated before task j; 0, otherwise
ws_m	Binary variable, 1, if workstation m is opened; 0, otherwise
l_i	Position number of task *i* in sequence

Based on the notations, entrance side sub-stations are coded with $[1,2,3, \ldots, M]$, and exit side sub-stations are coded with $[2M, 2M-1, 2M-2, \ldots, M+2, M+1]$.

Objectives and constrains:

$$\text{Min } F_1 = \sum_{m=1}^{M} ws_m \tag{1}$$

$$\text{Min } F_2 = \sum_{m=1}^{M} (CT - T_m)^2 \tag{2}$$

$$\text{Min } F_3 = \sum_{i=1}^{N} (s_i * h_i) \tag{3}$$

$$\text{Min } F_4 = \sum_{i=1}^{N} (s_i * d_i) \tag{4}$$

$$\sum_{m=1}^{M} \left(x_{im} + x_{i,2M+1-m}\right) = 1 \tag{5}$$

$$\sum_{i=1}^{N} \left(x_{im} + x_{i,2M+1-m}\right) \geq 1 \tag{6}$$

$$CT \geq T_m \tag{7}$$

$$x_{im} \leq \sum_{n=1}^{m} x_{jn} \quad \forall i, m;\ \forall j \in ANDP(i) \tag{8}$$

$$x_{im} \leq \sum_{j \in ORP(i)} \sum_{n=1}^{m} x_{jn} \quad \forall m, \forall i \in ORPT \tag{9}$$

$$w_{ii} = 0\ \forall i \tag{10}$$

$$w_{ij} + w_{ji} = 1 \quad \forall i, j \text{ and } i < j \tag{11}$$

$$T_m = \sum_{i=1}^{N} t_i \times \left(x_{im} + x_{i,2M+1-m}\right) + \sum_{i=1}^{N} \sum_{j=1}^{N} sd_{ji} \times (x'_{imj} + x'_{i,2M+1-m,j}) \quad \forall i, j \tag{12}$$

$$l_i = N - \sum_{j=1}^{N} w_{ij}\ \forall i \tag{13}$$

In this model, Eq. (1) introduces the first objective, minimizing number of workstations, which is a cost-based optimization criterion. Eq. (2) optimizes line smoothness, and this is a non-linear consideration. Eq. (3) indicates that hazardous parts should be removed early which aims to add environmental protection consciousness into

DLBP. Eq. (4) ensures that high demand parts are removed early which is a profit aspect objective. Constraint (5) ensures that one task can only be allocated to one sub-station. Constraint (6) shows that one workstation can operate one or more tasks. Constrain (7) is the cycle time constraint which avoids the situation of line stoppage. Constraint (8) and (9) seek to solve different precedence relationship. These two constraints ensure that the proposed MINLP model has the ability to solving complex precedence relationship. Constraint (10) and (11) present the sequence order of task i and j. Constraint (12) and (13) introduce the calculation process of total task processing time of workstation m and position in sequence of task i.

IV. Fish school search optimization (FSS)

FSS algorithm was originally proposed in research Bastos et al. (2008)[33] with the goal of minimizing searching time. FSS algorithm is inspired by the social behavior of fish schools, and it belongs to meta-heuristic algorithm. Three important operators in FSS algorithm are feeding, swimming, and breeding and these offer FSS algorithm wide-range search ability and great capability to switch between exploration and exploitation[34]. Considering the above-mentioned behaviors, the steps of implementing FSS can be summarized as follows:
Step 1: Initializing locations (x_i) randomly for all fish, setting all weights (w_i) to one.
Step 2: Starting the repeat loop.
Step 3: Performing swimming 1. Calculating random individual movement for each individual fish.
Step 4: Executing feeding operation. Updating weights for all fish based on new locations.
Step 5: Performing swimming 2. Collectively instinctive moving towards overall direction.
Step 6: Performing swimming 3. Collectively volitive moving dilation or contraction.
Step 7: Checking the termination condition.

4.1 Encoding and decoding

The FSS algorithm uses the same rule of task permutation for encoding in research (Kalayci and Gupta, 2013a, 2013b, 2013c, 2014; Li, Kucukkoc, and Zhang, 2019). Table 1 and Fig. 1 present related information of an encoding example for 8-part PC instance which is acquired from Kalayci and Gupta (2013a). It is clear from Fig. 1 that, lines between task 2 and 3, and task 5 and 6 are dashed, this means sequence-dependent situations are exist between related tasks. Sequence dependencies of this instance are provided as follows: $sd_{23} = 2, sd_{32} = 4, sd_{56} = 1, sd_{65} = 3$. Fig. 2 presents a feasible task assignment of 8-part small-size problem. In Fig. 2, it is clear that workstations are divided into sub-stations and each sub-station operates different tasks. The task permutation in Fig. 2 is 1, 2, 3, 4, 7, 8, 5, and 6, but the task sequence for objective calculation is 1, 2, 3, 6, 5, 8, 7, and 4. Difference between task permutation and task sequence should be noticed and an effective decoding algorithm is proposed to help transfer encoding into a feasible solution.

Table 1. Instance information for 8-part case

Task number	Part name	Task removal time	Hazardous index	Demand
1	PC top cover	14	No	360
2	Floppy drive	10	No	500
3	Hard drive	12	No	620
4	Back plane	18	No	480
5	PCI cards	23	No	540
6	RAM modules	16	No	750
7	Power supply	20	No	295
8	Motherboard	36	No	720

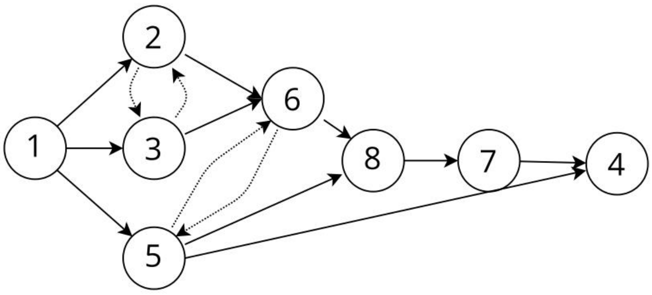

Fig. 1 Precedence relationship among 8 tasks

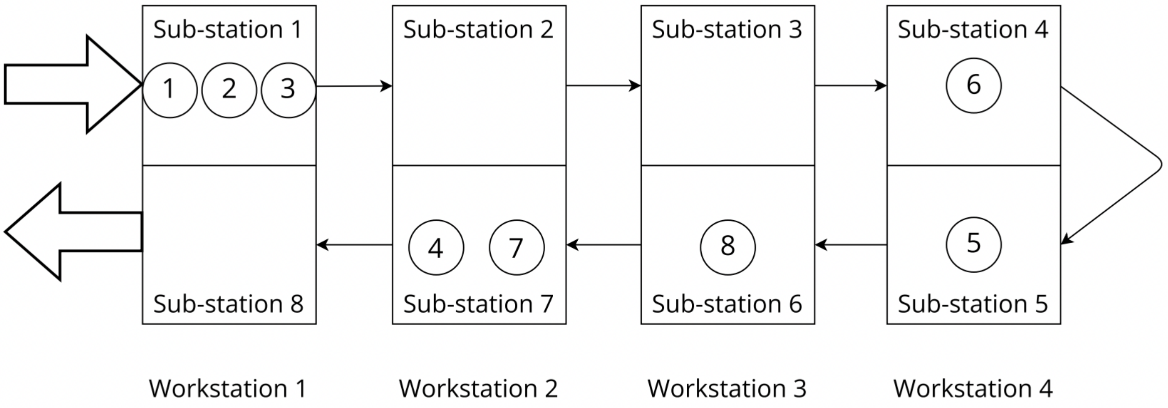

Fig. 2 Task assignment on a U-shaped line for 8-part PC instance

Table 2 and Table 3 present detailed information used in calculation of objective functions. For example, sub-station 1 disassembles task 1, 2 and 3, and with the consideration of sequence-dependent situation between task 2 and 3, the task processing time for this sub-station is 14+10+4+12=40. Since sub-station 8 has no task to remove, the total task processing time for workstation 1 is 40 and idle time is 0. For the first objective, there are 4 workstations and total idle times is 20. The objective value for removing hazardous part early is 0 since there is no hazardous task in this instance. For removing high demand parts early, the total amount of this objective is 19145.

Table 2. Task assignment and basic factors

Workstation number	Sub-station number	Task number	Task processing time	Total task processing time	Idle time
Workstation 1	Sub-station 1	1,2,3	14,10+4,12	40	0
	Sub-station 8	-	-		
Workstation 2	Sub-station 2	-	-	38	2
	Sub-station 7	4,7	18,20		
Workstation 3	Sub-station 3	-	-	34	4
	Sub-station 6	8	36		
Workstation 4	Sub-station 4	6	16+1	40	0
	Sub-station 5	5	23		

Table 3. Objective values

Objective number	Objective value
F_1	4
F_2	$0 + 2^2 + 4^2 + 0 = 20$
F_3	0 (No hazardous task)
F_4	$1 \times 360 + 2 \times 500 + 3 \times 620 + 4 \times 750 + 5 \times 540 + 6 \times 720 + 7 \times 295 + 8 \times 480 = 19145$

As mentation above, decoding procedure for SUDLBP is presented in Algorithm 1. Traditional decoding procedure for SDLBP is not suitable for SUDLBP since entrance and exit sides should be considered. One similar rule for both SDLBP and SUDLBP is high priority task should be assigned first. Algorithm 1 is presented as follows:

```
Algorithm 1. Decoding procedure
Start:
Step 1: Check termination criterion. If all tasks are assigned, terminate procedure; otherwise,  execute step 2.
Step 2: Open a new workstation.
Step 3: Add task(s), whose predecessor(s) has been assigned to the entrance side, to the available task set $A_{en}$;
        Add task(s), whose successor(s) has been assigned to the exit side, to the available task set $A_{ex}$.
Step 4: Add the task in $A_{en}$to the assignable task set $AS_{en}$ on the entrance side with the domain of cycle time
        constraint; Add the task in $A_{ex}$ to the assignable task set $AS_{ex}$ on the exit side with the domain of cycle
        time constraint. % For an assignable task, it can be assigned only the total task processing time of this
```

```
        workstation is less than or equal to the given cycle time with the considering of sequence dependency.
Step 5: If both two assignable task sets $AS_{en}$ and $AS_{ex}$ are empty, go back to step 1; otherwise, execute step 6.
Step 6: Select the task with higher priority of task permutation and allocate it to the entrance or exit side based
        on the situation; go back to step 3.
End
```

V. Case study and comparative study

Performance of FSS algorithm is provided in this section. FSS algorithm was first applied on two small-size instances, which are acquired from research Kalayci and Gupta (2013a). Information of EOL product and precedence relationship are presented in Table 4, Table 5, Fig. 3, and Fig. 4 respectively. The first instance has 10 parts, referred as P10, sequence dependencies of P10 instance are as follows: $sd_{1,4} = 1$, $sd_{4,1} = 4$, $sd_{2,3} = 2$, $sd_{3,2} = 3$, $sd_{4,5} = 4$, $sd_{5,4} = 2$, $sd_{5,6} = 2$, $sd_{6,5} = 4$, $sd_{6,9} = 3$, and $sd_{9,6} = 1$. The second instance contains 25 parts, referred as P25, sequence dependencies of P25 are $sd_{4,5} = 2$, $sd_{5,4} = 1$, $sd_{6,7} = 1$, $sd_{7,6} = 2$, $sd_{6,9} = 2$, $sd_{9,6} = 1$, $sd_{7,8} = 1$, $sd_{8,7} = 2$, $sd_{13,14} = 1$, $sd_{14,13} = 2$, $sd_{14,15} = 2$, $sd_{15,14} = 1$, $sd_{20,21} = 1$, $sd_{21,20} = 2$, $sd_{22,25} = 1$, and $sd_{25,22} = 2$. Cycle time for P10 and P25 is 40 and 18 respectively. To test the performance of FSS algorithm on large-size problems, an instance set is utilized which contains 47 different scale problems.

5.1 Case study

Table 4. 10-part product information

Task number	Part removal time	Hazardous index	Demand
1	14	No	0
2	10	No	500
3	12	No	0
4	17	No	0
5	23	No	0
6	14	No	750
7	19	Yes	295
8	36	No	0
9	14	No	360
10	10	No	0

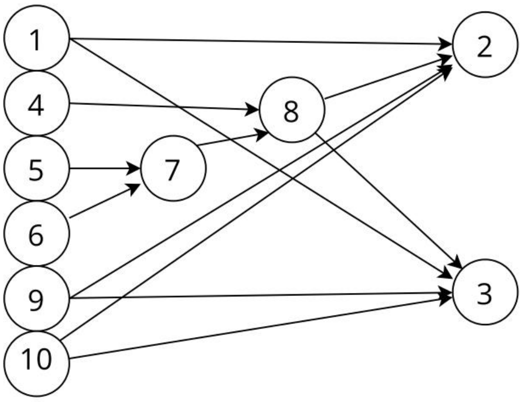

Fig. 3 Precedence relationship of P10

Table 5. 25-part product information

Task number	Part name	Part removal time	Hazardous index	Demand value
1	Antenna	3	1	4
2	Battery	2	1	7
3	Antenna guide	3	0	1
4	Bolt (Type 1) A	10	0	1
5	Bolt (Type 1) B	10	0	1
6	Bolt (Type 2) 1	15	0	1
7	Bolt (Type 2) 2	15	0	1
8	Bolt (Type 2) 3	15	0	1
9	Bolt (Type 2) 4	15	0	1
10	Clip	2	0	2
11	Rubber Seal	2	0	1

12	Speaker	2	1	4
13	White Cable	2	0	1
14	Red/Blue Cable	2	0	1
15	Orange Cable	2	0	1
16	Metal Top	2	0	1
17	Front Cover	2	0	2
18	Back Cover	3	0	2
19	Circuit Board	18	1	8
20	Plastic Screen	5	0	1
21	Keyboard	1	0	4
22	LCD	5	0	6
23	Sub-keyboard	15	1	7
24	Internal IC Board	2	0	1
25	Microphone	2	1	4

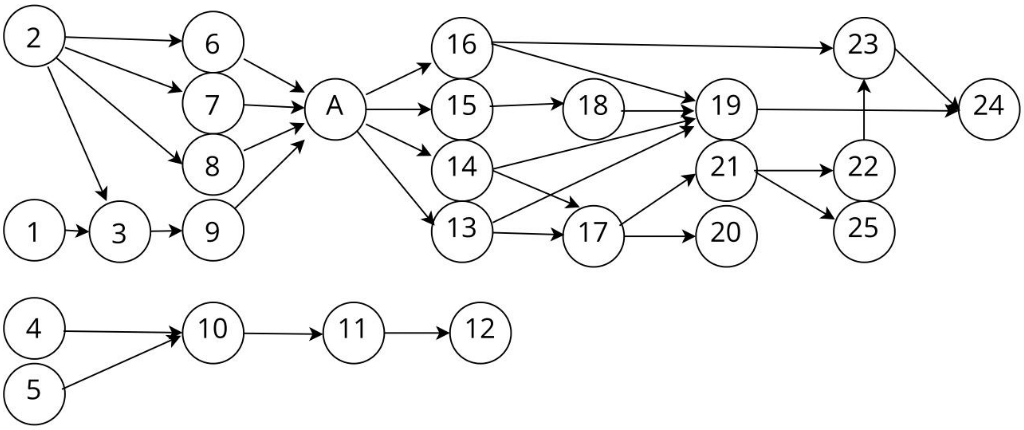

Fig. 4 Precedence relationship of P25

FSS algorithm is applied on a straight-line and a U-shaped line 20 times separately. Table 6 and Table 7 present objective values of P10 and P25 respectively. It is clear that, U-shaped line improves line efficiency and smoothness in cases studies. Also, compared with an iterated local search method (ILS) in research Li, Kuckkkoc, and Zhang (2019), FSS algorithm got the same four best values against ILS in terms of P10, and FSS has two better and two same best values against ILS in terms of P10. This may conclude that FSS algorithm is able to use on SDLBP and SUDLBP.

Table 6. Objective values of P10

Line type	Algorithm	Evaluation	F_1	F_2	F_3	F_4
SDLBP	FSS	Best value	5	67	5	9605
		Avg. value	5.00	67.00	5.00	9605.00
		S. D	0.00	0.00	0.00	0.00
SUDLBP	FSS	Best value	5	61	6	8880
		Avg. value	5.00	61.00	6.00	8880.00
		S. D	0.00	0.00	0.00	0.00

Table 7. Objective values of P25

Line type	Algorithm	Evaluation	F_1	F_2	F_3	F_4
SDLBP	FSS	Best value	10	9	80	925
		Avg. value	10.00	9.00	80.00	925.00
		S. D	0.00	0.00	0.00	0.00
SUDLBP	FSS	Best value	10	9	75	873
		Avg. value	10.00	9.00	78.35	894.65
		S. D	0.00	0.00	3.25	15.94

5.2 Comparative study

In this section, FSS algorithm is first compared with a genetic algorithm combined with variable neighborhood search method (VNSGA)[35] and iterated local search approach (ILS) (Li, Kucukkoc, and Zhang, 2019) for SDLBP and SUDLBP. Consideration of hierarchy method, only the best value of F_1 and F_2 are compared. F_1 has highest priority and F_2 has second highest priority. Results of VNSGA and ILS are taken from above mentioned research directly.

Table 8. Comparation between VNSGA, ILS, and FSS

Instance	N	CT	VNSGA (SDLBP)		ILS (SDLBP)		FSS (SDLBP)		ILS (SUDLBP)		FSS (SUDLBP)	
			F_1	F_2	F_1	F_2	F_1	F_2	F_1	F_2	F_1	F_2
Mertens	7	7	5	10	5	10	5	10	5	10	5	10
Bowman	8	20	5	149	5	149	5	149	4	13	4	13
Jaeschke	9	7	7	26	7	28	7	28	7	28	7	26
Jackson	11	10	5	6	5	6	5	6	5	4	5	4
Mansoor	11	94	2	5	2	5	2	5	2	5	2	5
Mitchell	21	15	8	31	8	43	8	31	8	29	8	29
Roszieg	25	16	8	5	8	5	8	5	8	3	8	3
Heskiaoff	28	216	5	628	5	630	5	628	5	628	5	628
Buxey	29	30	12	118	12	122	12	122	11	6	11	6
Lutzl	32	2357	7	8.13E+05	7	8.47E+05	7	8.49E+05	7	7.99E+05	7	8.03E+05
Gunther	35	41	14	1519	14	1735	14	1779	12	13	12	13
Kilbridge	45	62	9	6	9	6	9	6	9	6	9	6
Hahn	53	2806	6	1.87E+06	6	1.91E+06	6	1.87E+06	5	6	5	6
Tonge	70	168	22	2152	22	1756	22	1874	22	1672	22	1708
Tonge	70	170	22	3002	22	2660	22	2892	21	204	21	780
Tonge	70	173	22	5196	21	1081	21	1625	21	745	21	823
Tonge	70	179	21	3459	20	312	20	518	20	262	20	288
Tonge	70	182	20	968	20	912	20	952	20	854	20	880
Wee-Mag	75	46	35	983	34	399	34	393	34	349	34	345
Wee-Mag	75	47	33	148	33	116	33	120	33	106	33	112
Wee-Mag	75	49	32	189	32	163	32	163	32	155	32	157
Wee-Mag	75	50	32	347	32	333	32	331	32	327	32	325
Wee-Mag	75	52	31	455	31	443	31	449	31	431	31	439
Arcus1	83	3985	20	9.34E+05	20	9.22E+05	20	9.18E+05	20	8.14E+05	20	8.14E+05
Arcus1	83	5048	16	1.76E+06	16	1.76E+06	16	1.74E+06	16	1.67E+06	16	1.66E+06
Arcus1	83	5853	14	2.79E+06	14	2.79E+06	14	2.79E+06	13	1.16E+04	13	1.30E+04
Arcus1	83	6842	12	4.26E+06	12	4.25E+06	12	4.24E+06	12	3.43E+06	12	3.42E+05
Arcus1	83	7571	11	5.37E+06	11	5.54E+06	11	5.36E+06	11	5.37E+06	11	5.36E+06
Arcus1	83	8412	10	7.09E+06	10	7.83E+06	10	7.25E+06	10	7.93E+06	10	7.19E+06
Arcus1	83	8898	9	2.14E+06	9	2.15E+06	9	2.14E+06	9	2.13E+06	9	2.13E+06
Arcus1	83	10816	8	1.49E+07	8	3.75E+07	8	1.59E+07	7	1.10E+01	7	1.08E+07
Lutz2	89	15	34	63	34	61	34	61	33	10	33	10
Lutz3	89	150	12	2050	12	2256	12	2178	11	6	11	6
Mukherjee	94	201	23	12057	23	14853	23	13475	21	13	21	13
Mukherjee	94	301	15	10137	15	10137	15	10137	14	6	14	14
Arcus2	111	5755	27	2.58E+06	27	2.40E+06	27	2.44E+06	27	1.06E+06	27	1.04E+06
Arcus2	111	7520	21	3.00E+06	21	2.97E+06	21	2.93E+06	21	2.75E+06	21	2.72E+06
Arcus2	111	8847	18	4.38E+06	18	4.59E+06	18	4.40E+06	18	4.41E+06	18	4.39E+06
Arcus2	111	10027	16	6.33E+06	16	6.39E+06	16	6.35E+06	16	6.42E+06	16	6.33E+06
Arcus2	111	10743	15	7.76E+06	15	7.82E+06	15	7.76E+06	15	7.81E+06	15	7.76E+06
Arcus2	111	11378	14	5.76E+06	14	5.72E+06	14	5.70E+06	14	5.68E+06	14	5.64E+06
Arcus2	111	11570	14	9.86E+06	14	1.02E+07	14	9.98E+06	14	9.63E+06	14	9.59E+06
Arcus2	111	17067	9	1.14E+06	9	1.14E+06	9	1.14E+06	9	1.14E+06	9	1.14E+06
Barthol2	148	85	52	906	51	293	51	313	51	243	51	251
Barthol2	148	89	50	1174	49	425	49	673	48	74	48	103
Barthol2	148	91	49	1179	48	504	48	486	47	67	47	66
Barthol2	148	95	47	1279	46	454	46	450	45	53	45	53

Solutions in Table 8 are compared in two aspects, straight-line and U-shaped line comparation. On a straight-line layout, FSS has 7 better and 40 same values on F_1 and 22 better and 14 same values on F_2 against VNSGA. Also, considering of straight-line disassembly, FSS obtains 47 same values in terms of F_1 and 21 better and 13 same values in terms of F_2 against ILS. For SUDLBP, FSS is only compared with ILS on F_1 and F_2. Results of best objective value found present that FSS has 47 out of 47 same values on F_1, and 16 better and 17 same values on F_2. It can be concluded that, FSS algorithm has the ability of finding optimal solutions for sequence-dependent U-shaped DLBP. Results of U-shaped layout are much lower than that of straight-line configuration, which indicates that U-shaped layout improves productivity and line smoothness against straight-line configuration. To test the algorithm searching ability on SUDLBP, FSS algorithm is compared with 9 other algorithms which include hill-climbing algorithm (HC) (McGovern and Gupta, 2007a, 2007b), late acceptance hill-climbing algorithm (LAHC)[36], simulated annealing algorithm (SA), tabu search algorithm (TS), genetic algorithm (GA), artificial bee colony algorithm (ABC), bee algorithm (BA), particle swarm optimization (PSO), and iterated local search optimization (ILS). Results of average objective values on F_1 and F_2 are presented in Table 9 and Table 10 respectively. Notice that SA, TS, GA, ABC, BA, and PSO are re-implemented 20 times each on a U-shaped line, and results of HC, LAHC, and ILS are acquired from research Li, Kucukkoc, and Zhang (2019).

From Table 9, FSS algorithm gets 43 out of 47 the most optimal solutions. Also, from Table 10, FSS obtains 31 out of 47 best solutions compared with 9 other meta-heuristic algorithms. Results can be concluded that, FSS algorithm is suitable for utilizing on SUDLBP and it has a superior performance.

Table 9. Comparation of average value in terms of F_1

Instance	N	CT	HC	LAHC	SA	TS	GA	ABC	BA	PSO	ILS	FSS
Mertens	7	7	5	5	5	5	5	5	5	5	5	5
Bowman	8	20	4	4	4	4	4	4	4	4	4	4
Jaeschke	9	7	7	7	7	7	7	7	7	7	7	7
Jackson	11	10	5	5	5	5	5	5	5	5	5	5
Mansoor	11	94	2	2	2	2	2	2	2	2	2	2
Mitchell	21	15	8	8	8	8	8	8	8	8	8	8
Roszieg	25	16	8	8	8	8	8	8	8	8	8	8
Heskiaoff	28	216	5	5	5	5	5	5	5	5	5	5
Buxey	29	30	11	11.05	11	11	11	11	11	11	11	11
Lutzl	32	2357	7	7	7	7	7	7	7	7	7	7
Gunther	35	41	12	12	12	12	12	12	12	12.15	12	12
Kilbridge	45	62	9	9	9	9	9	9	9	9	9	9
Hahn	53	2806	5.7	5.65	5.6	5.85	5.9	6.0	5.5	5.5	5.2	5.25
Tonge	70	168	22	22	22.15	22	22	22	22	22	22	22
Tonge	70	170	21.95	21.95	22.00	21.95	21.8	22.00	22.00	21.75	21.8	21.9
Tonge	70	173	21	21	21	21	21	21	21	21.15	21	21
Tonge	70	179	20	20	20	20	20	20	20	20	20	20
Tonge	70	182	20	20	20	20	20	20	20	20	20	20
Wee-Mag	75	46	34	34	34	34.3	34	34.9	34	34	34	34
Wee-Mag	75	47	33	33	33	33	33	33	33	33	33	33
Wee-Mag	75	49	32	32	32	32	32	32	32	32	32	32
Wee-Mag	75	50	32	32	32	32	32	32.1	32	32	32	32
Wee-Mag	75	52	31	31	31	31	31	31	31	31	31	31
Arcus1	83	3985	20	20	20.05	20	20	20	20.05	20	20	20
Arcus1	83	5048	16	16	16	16	16	16	16	16	16	16
Arcus1	83	5853	13	13	13.4	13	13.75	13.5	13.5	13	13	13
Arcus1	83	6842	12	12	12	12	12	12	12	12	12	12
Arcus1	83	7571	11	11	11	11	11	11	11	11	11	11
Arcus1	83	8412	10	10	10	10	10	10	10	10	10	10
Arcus1	83	8898	9	9	9	9	9	9	9	9	9	9
Arcus1	83	10816	8	8	8	8	8	8	8	8	7.8	7.9
Lutz2	89	15	33	33	33.25	33.25	33.15	33.25	33	33	33	33
Lutz3	89	150	11	11	11.25	11	11	11	11	11.25	11	11
Mukherjee	94	201	21.25	21.2	21.5	21.95	21.5	22.00	21.5	21.85	21.25	21.2
Mukherjee	94	301	14	14	14.5	14.7	14.75	15.0	14.7	14	14	14
Arcus2	111	5755	27	27	27	27	27	27	27	27	27	27
Arcus2	111	7520	21	21	21	21	21	21	21	21	21	21
Arcus2	111	8847	18	18	18	18	18	18	18	18	18	18
Arcus2	111	10027	16	16	16	16	16	16	16	16	16	16
Arcus2	111	10743	15	15	15	15	15	15	15	15	15	15
Arcus2	111	11378	14	14	14	14	14	14	14	14	14	14
Arcus2	111	11570	14	14	14	14	14	14	14	14	14	14
Arcus2	111	17067	9	9	9	9	9	9	9	9	9	9
Barthol2	148	85	51	51	51.3	51.7	51.75	51.3	51	51.15	51	51
Barthol2	148	89	49	48.9	48.95	48.95	49.00	49.00	48.95	49.05	48.75	48.9
Barthol2	148	91	48	47.8	48.0	48.0	48.1	47.9	47.8	48.0	47.6	47.5
Barthol2	148	95	45.9	45.85	46.00	45.95	45.75	46.00	45.85	46.00	45.65	45.6

Table 10. Comparation of average value in terms of F_2

Instance	N	CT	HC	LAHC	SA	TS	GA	ABC	BA	PSO	ILS	IWO
Mertens	7	7	10	10	10	10	10	10	10	10	10	10
Bowman	8	20	13	13	13	13	13	13	13	13	13	13
Jaeschke	9	7	28	28	28	28	28	28	28	28	28	28
Jackson	11	10	4	4	4	4	4	4	4	4	4	4
Mansoor	11	94	5	5	5	5	5	5	5	5	5	5
Mitchell	21	15	30.7	31	29.3	29.8	30.0	29.7	30.1	29.8	29.1	29.2
Roszieg	25	16	3.2	3.9	3	3	3	3	3.1	3	3	3
Heskiaoff	28	216	634.8	636.4	629.9	630.8	629.9	630.7	629.1	629.2	629.1	628.9
Buxey	29	30	8.4	15.8	6.9	7.3	8.4	7.1	6.8	6.4	6.5	6.4
Lutzl	32	2357	838157	830279	827934	825674	817453	819426	845392	835712	804475	812942

Gunther	35	41	13	13.4	13.9	13.6	14.0	13.5	13.6	13.3	13.1	13.1
Kilbridge	45	62	6.2	8.9	6	6.3	6.3	6	6	6	6	6
Hahn	53	2806	1E+06	1E+06	1E+06	976892	992605	966721	1E+06	1E+06	344411	312974
Tonge	70	168	1805.5	1811.3	1922.3	1873.2	1799.3	1869.3	1801.7	1786.1	1783.0	1809.4
Tonge	70	170	2690.9	2651.8	2894.1	2605.3	2871.5	2973.4	3105.5	2734.8	2159.8	2579.5
Tonge	70	173	1088.8	1719.7	1105.4	1089.1	891.4	1005.5	995.1	915..5	954.1	913.6
Tonge	70	179	325.6	518.5	395.7	401.5	298.4	341.2	318.1	286.5	290.8	342.5
Tonge	70	182	934	1685.7	939.0	954.2	907.8	892.9	1091.5	972.4	879.9	897.6
Wee-Mag	75	46	475.4	457.5	397.9	418.2	482.3	501.5	472.3	412.3	426.7	370.9
Wee-Mag	75	47	128.5	118.0	118.5	121.4	123.5	120.9	119.3	119.9	117.3	118.3
Wee-Mag	75	49	159.9	159.5	160.2	167.3	159.3	164.5	157.5	158.9	159.3	159.7
Wee-Mag	75	50	337.8	331.5	335.9	329.8	333.5	340.2	329.1	332.6	330.5	328.7
Wee-Mag	75	52	446.9	444.4	449.3	447.1	439.9	450.3	439.2	437.9	437.8	442.7
Arcus1	83	3985	838896	835347	847250	860971	841965	864210	834702	830739	827898	820583
Arcus1	83	5048	2E+06	2E+06	2E+06	2E+06	2E+06	2E+06	2E+06	2E+06	2E+06	2E+06
Arcus1	83	5853	13515	19389	1E+06	1E+06	50542	77934	1E+06	19245	12786	14508
Arcus1	83	6842	4E+06	4E+06	3E+06	3E+06	3E+06	4E+06	4E+06	4E+06	4E+06	4E+06
Arcus1	83	7571	6E+06	6E+06	6E+06	7E+06	7E+06	6E+06	6E+06	6E+06	6E+06	6E+06
Arcus1	83	8412	1E+07	1E+07	1E+07	1E+07	1E+07	9E+06	9E+06	9E+06	1E+07	9E+06
Arcus1	83	8898	2E+06	2E+06	2E+06	2E+06	2E+06	2E+06	2E+06	2E+06	2E+06	2E+06
Arcus1	83	10816	4E+07	4E+07	4E+07	4E+07	4E+07	4E+07	3E+07	4E+07	3E+07	3E+07
Lutz2	89	15.0	10.3	16.5	23.5	10.2	11.3	12.4	11.9	10.3	10.1	10.2
Lutz3	89	150	6.4	10.7	7.8	6.2	7.0	105.9	37.2	9.6	6.6	6.3
Mukherjee	94	201	588.25	475.1	307.5	962.4	1205.1	1024.7	923.4	678.5	564.35	297.8
Mukherjee	94	301	14.4	16.5	92.7	1125.6	784.3	2736.5	4892.6	19.2	9.6	15.8
Arcus2	111	5755	1E+06	2E+06	1E+06	1E+06	1E+06	1E+06	1E+06	1E+06	1E+06	1E+06
Arcus2	111	7520	3E+06	3E+06	3E+06	3E+06	3E+06	4E+06	3E+06	3E+06	3E+06	3E+06
Arcus2	111	8847	5E+06	5E+06	5E+06	6E+06	5E+06	5E+06	5E+06	5E+06	5E+06	5E+06
Arcus2	111	10027	7E+06	7E+06	7E+06	7E+06	7E+06	8E+06	7E+06	7E+06	7E+06	7E+06
Arcus2	111	10743	8E+06	8E+06	9E+06	9E+06	9E+06	9E+06	8E+06	9E+06	8E+06	8E+06
Arcus2	111	11378	6E+06	6E+06	6E+06	6E+06	6E+06	6E+06	6E+06	6E+06	6E+06	6E+06
Arcus2	111	11570	1E+07	1E+07	1E+07	1E+07	1E+07	1E+07	1E+07	1E+07	1E+07	1E+07
Arcus2	111	17067	1E+06	1E+06	1E+06	1E+06	1E+06	1E+06	1E+06	1E+06	1E+06	1E+06
Barthol2	148	85	259.8	258.4	295.6	338.4	329.6	673.4	901.7	260.8	257.4	293.5
Barthol2	148	89	371.2	346.0	378.2	396.4	417.2	409.5	553.1	384.7	294.65	341.9
Barthol2	148	91	414.0	362.4	392.4	525.3	387.0	615.6	425.3	382.6	281.3	275.5
Barthol2	148	95	419.4	396.95	435.0	392.5	624.8	592.5	371.3	917.3	311.65	323.6

VI. Conclusions

With the decreasing of rare-earth resources and increasing concerns of waste problem. Product recovery has become an important method in sustainability development since it properly deals with EOL products. Remanufacturing is a smart strategy which can solve part of waste and pollution problem especially waste of electrical and electronic equipment (WEEE). Disassembly is widely employed in industrial to transfer EOL products into different parts. These social background makes DLBP research area become attractive. With the consideration of finding near-optimal solutions in acceptable computation time, this paper has for the first time proposed FSS algorithm on sequence-dependent U-shaped DLBP and greatly expands the research field. A MINLP model with the ability of solving complex precedence relationships is implemented and tested on straight-line and U-shaped line separately. Four different objectives are introduced in this paper to test the performance of FSS algorithm on different aspects. Two instance sets which contain small-size and large-size cases are set as benchmark problems to test the performance of FSS algorithm. Results of case studies and comparative studies indicate that U-shaped layout can improves line efficiency and smoothness, and FSS algorithm is a great meta-heuristic algorithm which can be future utilized in DLBP.

In the future, to expand research field of DLBP, novel approaches can be applied on a disassembly line. Also, due to the limited number of studies of U-shaped DLBP and sequence-dependent DLBP, these two aspects are attractive to researchers. What is more, real industrial and commercial cases are welcomed to be introduced in DLBP, especially cases with uncertain task processing time and instances with OR precedence relationships.

References

[1] Yin, T., Zhang, Z., Zhang, Y., Wu, T. and Liang, W., 2022. Mixed-integer programming model and hybrid driving algorithm for multi-product partial disassembly line balancing problem with multi-robot workstations. *Robotics and Computer-Integrated Manufacturing*, *73*, p.102251.

[2] Zhang, L., Zhao, X., Ke, Q., Dong, W. and Zhong, Y., 2021. Disassembly line balancing optimization method for high efficiency and low carbon emission. *International Journal of Precision Engineering and Manufacturing-Green Technology*, *8*(1), pp.233-247.

[3] Siddig, N., Zhang, Z., Mokhtar, A. and Abualnor, A., 2021. Optimizing Disassembly Line Balancing Problem Using Multi-objective ADAM Optimizer Algorithm. *Journal of Karary University for Engineering and Science*.

[4] Gungor, A. and Gupta, S.M., 1999b. Issues in environmentally conscious manufacturing and product recovery: a survey. *Computers & Industrial Engineering*, 36(4), pp.811-853.

[5] Yao, P. and Gupta, S. M., 2021a. Cat Swarm Optimization Algorithm for Solving Multi-Objective U-Shaped Disassembly Line Balancing Problem. *Proceedings of the International Conference on Remanufacturing*, March 24-25, pp. 222-230.

[6] Hjorth, S. and Chrysostomou, D., 2022. Human–robot collaboration in industrial environments: A literature review on non-destructive disassembly. *Robotics and Computer-Integrated Manufacturing*, *73*, p.102208.

[7] Gungor, A., & Gupta, S. M., 1999a. Disassembly line balancing. *Proceedings of the 1999 Annual Meeting of the Northeast Decision Sciences Institute, Newport*, Rhode Island, March 24-26, pp.193-195.

[8] Yin, T., Zhang, Z. and Jiang, J., 2021. A Pareto-discrete hummingbird algorithm for partial sequence-dependent disassembly line balancing problem considering tool requirements. *Journal of Manufacturing Systems*, *60*, pp.406-428.

[9] Yao, P. and Gupta, S. M., 2021b. Small World Optimization Algorithm for Solving Multi-Objective U-Shaped Disassembly Line Balancing Problem, *Proceedings of the 2021 Annual Meeting of the Northeast Decision Sciences Institute*, Virtual, March 26-27, 659-668.

[10] Kalayci, C.B. and Gupta, S.M., 2013a. Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem. *Expert Systems with Applications*, *40*(18), pp.7231-7241.

[11] Yao, P. and Gupta, S. M., 2021c. Ant Colony Optimization Algorithm for Solving U-Shaped Disassembly Line Balancing Problem with Multiple Objectives, *Proceedings of the 4th International Conference on Innovative Studies of Contemporary Sciences*, Tokyo, Japan, July 29-31, pp. 21-26.

[12] McGovern, S.M. and Gupta, S.M., 2007a. A balancing method and genetic algorithm for disassembly line balancing. *European journal of operational research*, *179*(3), pp.692-708.

[13] McGovern, S.M. and Gupta, S.M., 2007b. Combinatorial optimization analysis of the unary NP-complete disassembly line balancing problem. *International Journal of Production Research*, *45*(18-19), pp.4485-4511.

[14] Özceylan, E., Kalayci, C.B., Güngör, A. and Gupta, S.M., 2019. Disassembly line balancing problem: a review of the state of the art and future directions. *International Journal of Production Research*, *57*(15-16), pp.4805-4827.

[15] Kalayci, C.B. and Gupta, S.M., 2013b. A particle swarm optimization algorithm with neighborhood-based mutation for sequence-dependent disassembly line balancing problem. *The International Journal of Advanced Manufacturing Technology*, *69*(1), pp.197-209.

[16] Kalayci, C.B. and Gupta, S.M., 2013c. Ant colony optimization for sequence- dependent disassembly line balancing problem. *Journal of Manufacturing Technology Management*.

[17] Kalayci, C.B. and Gupta, S.M., 2013d. Simulated annealing algorithm for solving sequence-dependent disassembly line balancing problem. *IFAC Proceedings Volumes*, *46*(9), pp.93-98.

[18] Kalayci, C.B. and Gupta, S.M., 2014. A tabu search algorithm for balancing a sequence-dependent disassembly line. *Production Planning & Control*, *25*(2), pp.149-160.

[19] Yao, P. and Gupta, S. M., 2021d. Invasive Weed Optimization Algorithm for Solving Multi-Objective U-Shaped Disassembly Line Balancing Problem", *Proceedings of the 12th International Congress on Mathematics, Engineering and Natural Sciences*, Paris, France, July 9-11, pp. 286-292.

[20] Yao, P. and Gupta, S. M., 2021e. Teaching-Learning-Based Optimization Algorithm for Solving Multi-Objective U-Shaped Disassembly Line Balancing Problem, *Proceedings of the 5th International New York Conference on Evolving Trends in Interdisciplinary Research and Practices*, Manhattan, New York City, October 3-5, pp. 21-28.

[21] Yao, P. and Gupta, S. M., 2021f. Fish School Search Algorithm for Solving Multi-Objective U-Shaped Disassembly Line Balancing Problem, *Proceedings of the Latin American International Conference on Natural and Applied Sciences*, Villahermosa, Mexico, November 5-6, pp. 44-52.

[22] Agrawal, S., and Tiwari, M.K., 2008. A collaborative ant colony algorithm to stochastic mixed-model U-shaped disassembly line balancing and sequencing problem. *International Journal of Production Research, 46*(6), pp.1405-1429.

[23] Avikal, S. and Mishra, P.K., 2012. A new U-shaped heuristic for disassembly line balancing problems. *PRATIBHA: International Journal of Science, Spirituality, Business and Technology, 1*(1), pp.21-27.

[24] Avikal, S., Jain, R. and Mishra, P., 2013. A heuristic for U-shaped disassembly line balancing problems. *MIT International Journal of Mechanical Engineering, 3*(1), pp.51-56.

[25] Wang, K., Gao, L. and Li, X., 2020. A multi-objective algorithm for U-shaped disassembly line balancing with partial destructive mode. *Neural Computing and Applications*, *32*(16), pp.12715-12736.

[26] Li, Z. and Janardhanan, M.N., 2021. Modelling and solving profit-oriented U-shaped partial disassembly line balancing problem. *Expert Systems with Applications*, *183*, p.115431.

[27] Wang, W., Guo, X., Liu, S., Qin, S.J., Qi, L., Zhao, Z. and Tang, Y., 2021, October. Multi-objective Discrete Chemical Reaction Optimization Algorithm for Multiple-product Partial U-shaped Disassembly Line Balancing Problem. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 2322-2327). IEEE.

[28] Wu, K., Guo, X., Liu, S., Qi, L., Zhao, J., Zhao, Z. and Wang, X., 2021, May. Multi-objective discrete brainstorming optimizer for multiple-product partial u-shaped disassembly line balancing problem. In *2021 33rd Chinese Control and Decision Conference (CCDC)* (pp. 305-310). IEEE.

[29] Lu, F., Liu, P., Qi, L., Qin, S., Xu, G. and Xu, Z., 2021, September. Multi-objective discrete strength pareto evolutionary algorithm II for multiple-product partial U-shaped disassembly line balancing problem. In *Journal of Physics: Conference Series* (Vol. 2024, No. 1, p. 012058). IOP Publishing.

[30] Li, Z., Kucukkoc, I. and Zhang, Z., 2019. Iterated local search method and mathematical model for sequence-dependent U-shaped disassembly line balancing problem. *Computers & Industrial Engineering*, *137*, p.106056.

[31] Li, Z., Kucukkoc, I., Tang, Q. and Zhang, Z., 2021. Models and two-phase bee algorithms for multi-objective U-shaped disassembly line balancing problem. *Optimization and Engineering*, pp.1-32.

[32] Wang, Y., Xie, Y., Ren, Y. and Zhang, C., 2021, February. A MCDM-Based Meta-Heuristic Approach for U-shaped Disassembly Line Balancing Problem. In *Journal of Physics: Conference Series* (Vol. 1828, No. 1, p. 012159). IOP Publishing.

[33] Bastos Filho, C.J., de Lima Neto, F.B., Lins, A.J., Nascimento, A.I. and Lima, M.P., 2008, October. A novel search algorithm based on fish school behavior. In *2008 IEEE international conference on systems, man and cybernetics* (pp. 2646-2651). IEEE.

[34] Bastos Filho, C.J., de Lima Neto, F.B., da CC Lins, A.J., Nascimento, A.I. and Lima, M.P., 2009. Fish School Search. *Nature-inspired algorithms for optimisation*, *193*, pp.261-277.

[35] Kalayci, C.B., Polat, O. and Gupta, S.M., 2016. A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem. *Annals of Operations Research*, *242*(2), pp.321-354.

[36] Yuan, B., Zhang, C. and Shao, X., 2015. A late acceptance hill-climbing algorithm for balancing two-sided assembly lines with multiple constraints. *Journal of Intelligent Manufacturing*, *26*(1), pp.159-168.

Pengfei Yao, et. al. "Fish School Search Optimization Algorithm for Solving U-Shaped Sequence-Dependent Disassembly Line Balancing Problem with Multiple Objectives." *IOSR Journal of Engineering (IOSRJEN)*, 12(06), 2022, pp. 10-21.