



A Novel Recursive Heuristic Algorithm for Straight and U-Shaped Assembly Line Balancing with Fuzzy Processing Time

Samah. A. Aufy ^{a*}, AllaEldin. H. Kassam  ^b

^a Production Engineering and Metallurgy Department, University of Technology, Baghdad, Iraq,
70221@uotechnology.edu.iq.

^b Production Engineering and Metallurgy Department, University of Technology, Baghdad, Iraq,
70150@uotechnology.edu.iq.

*Corresponding author.

Submitted: 10/09/2020

Accepted: 07/12/2020

Published: 25/03/2021

KEYWORDS

Assembly line, Fuzzy processing time, Recursive Heuristic algorithm, Cycle time

ABSTRACT

The paper aims to address the straight and U-type assembly line balancing problems by developing a novel recursive heuristic algorithm based on the idea of the depth of search. The dynamic fuzzy processing time (DFPT) model is employed to represent uncertainty and ambiguity related to the processing time in the actual production systems. The novel algorithm, the minimum cycle time objective is considered for a set of imposed considerations. They are arranged in an appropriate strategy in which three-stages are proposed and presented as a solution approach. Finally, the validity of the developed solution approach is evaluated through a tested numerical example conducted over a test problem taken from literature to assess its performance. This study proves their ability and efficiency in assisting decision-making by determining the contribution proportion for significant assignment variables represented by skill level, work stability, type layout, and priority rule.

How to cite this article:, "A Novel Recursive Heuristic Algorithm for Straight and U-Shaped Assembly Line Balancing with Fuzzy Processing Time," Engineering and Technology Journal, Vol. 39, Part A, No. 03, pp. 477-487, 2021.

DOI: <https://doi.org/10.30684/etj.v39i3A.1830>

This is an open access article under the CC BY 4.0 license <http://creativecommons.org/licenses/by/4.0>

1. INTRODUCTION

The growing competitive market leads manufacturing organizations to engage themselves in productivity improvement plan aiming at more efficient and effective plans in shorting time [1,2]. In this direction, by Henry Ford the manufacturing assembly line was presented for structuring an efficient assembly line is considered as one of the most essential in enhancing an assembly line [1,3]. In general, an assembly line is an organized way used to produce a high volume of standard products on a sequence of workstations ($S=1,2, \dots, m$) ordered along a conveyor belt or by another material handling system [4]. A working mechanism of assembly line balancing problem (ALBP) based on consecutively assembled moved from one workstation into its successive until complete all tasks

required to produce a final product with respect to the precedence relations among assigned tasks as well other imposed constraints [1,4]. In the literature, there are numerous studies about ALBP usually classified into two basic categories representing a simple assembly line balancing problem (SALBP) and generalized assembly line balancing problem (GALBP). However, all versions of SALBP are divided into their objective [5,6]. This work is related to only SALBP-type I and SALBP-type II versions have a dual relation while the first type tries to minimize the number of workstations for given cycle time, the second one tries to minimize the cycle time for given workstations number [7,8]. GALBP versions have the extra features included other classification depending on its layout into two group straight assembly line (S-ALB) and U-shaped assembly line (U-ALB) [8]. S-ALB can be considered as one important conventional mass production system, while recently, because of continuous improvement and cost reduction, U-ALB are presented [9]. However, in this study, another feature under human operator is addressed despite technological improvement in manufacturing it is still often referred to as a high level of automation system because they are the most efficient solution [10]. Even so, the work processing time of the assembly line usually has been considered smooth, in another word it is performed in standard time but, this can lead to delays about estimating dates to complete product [11]. But that is accrued reality due to a high degree of variability, ambiguity and vagueness are experienced as a result from worker stress, unskilled workers, employee interests, lack of training, etc., so not meet demand may result from machine breakdowns. Hence, task processing time can be handled by estimating uncertain data to integrate process time uncertainty in ALBP [12].

2. RELATED STUDIES

Assembly line balancing problem has become a major concern of academic literary researchers. There are many numerous heuristic approaches, exact algorithms, and methods of optimization that were developed to solve the problem under investigation. Nevertheless, most of the literature studies concentrated on the heuristic approach and model type and so on.

Yakupk et al. (2009) [13] formulated a binary fuzzy goal programming solution approach for solving ALBP. This approach is presented to balance straight and U-shaped assembly systems with fuzzy while the number of workstations and cycle time is considered as the goal. A numerical example was to validate the developed approach. All studies prove that the development of a fast and efficient heuristic method. Uğur Ö. and Bilal T. (2009) [4] presented an adaptive learning approach, and simulated annealing as a proposed approach. It is addressed simple straight and U-shaped ALBP. Validation of the performance was conducted through solving many benchmark problems available in the literature. Finally, the obtained result of the experimental study proofs the superiority of the proposed approach overall examined problems as well as their optimal solutions which were captured in short computational time. Shwetank A. et al. (2013) [14] presented an approach based on a critical path method (CPM) and evaluate labor productivity in straight and U-shape assembly line systems. The obtained results show the proposed approach based U-shaped gives better solutions in improving productivity by CPM based straight approach. Alavidoe M. et al. (2016) [1] introduced a novel model represented by bi-objective fuzzy mixed-integer linear programming (BOFMILP) model to handle straight and U-shaped ALBP in an uncertain environment. The proposed algorithm is proved its ability to address any practical multi-objective linear programming model in assisting decision-makers to deal with an uncertainty environment. Hadi P. T. et al. (2017) [15] formulated cost and cycle time objectives decision-making problem as a mixed-integer nonlinear optimization for solving simple assembly line. Multi-objective particle swarm optimization (MOPSO) and hybrid method resulted in more non-dominated Pareto solutions to evaluate this model. The primary yielded proof MOPSO has provided a better-quality solution. Domain K. and Karol D. (2020) [16] presented a proposal of a new hybrid heuristic algorithm that integrated a modified ranking positional weight with a local search of task order on assembly workstation zones. The obtained results of the experimental study indicated the effectiveness and reliability of the proposal. Aufy and Kassam (2020) [17] presented a proposal for a new methodology for balancing a mixed-model assembly line using a worker-assigned heuristic workstation (W-TAWH) model to handle straight and U-shaped problems. The proposal enhanced performance measures depending on the number of suitable workers and tasks that assigned to the given workstation. Finally, these measures are integrated and optimized by employing the desirability function approach for optimization. This paper presented consecutive and recursive heuristic algorithms with the main objective is minimum cycle time for

solving both S-ALB and U-ALB in a first attempts to study the effect of design considerations represented solution approach, layout, sequence vector, and the number of workstations simultaneously in re-balancing assembly line.

3. MATHEMATICAL FORMULATION

The paper is presented an extension study related to balance a mixed model assembly line. It is formulated in worker–tasks stochastic assigned into given workstations (W-TSAGW) model concerns with assigning a proper tasks and proper worker to a given workstation to minimize cycle time under the constraint of precedence relations. This study considers the developed model subjected to the set of assumptions which are:

1. The task must be assigned to available workstations if the preceding relationship has not been violated.
2. Stochastic processing time of the tasks is considered.
3. The sum of processing times allocated to each workstation must not exceed the cycle time.
4. Given the number of workstations, the worker number must be identical (it ensures how every worker is assigned to only one workstation).

Indices

i: index of task ($i=1,2,\dots,n$)

k: index of worker ($k=1,2,\dots,m$)

j: index of product ($j=1,2,\dots,p$)

Parameters

n= number of tasks

m= number of workers

S= number of workstations

P= number of products

$P_{ts(max)}$ =max total number of successor tasks

$P_{tp(min)}$ = min total number of predecessor tasks

$P_{maxf(c)}$ = max total number of successor or predecessor tasks

$P_{minf(c)}$ = min total number of successor or predecessor tasks

μ_c^p = set of tasks (*p*) that precede task

μ_c^s = set of tasks (*s*) which succeed task

Decision Variable

$$A_{si} = \begin{cases} 1, & \text{if task } i \text{ assigned to workstation } s \\ 0, & \text{otherwise} \end{cases}$$

4. MATHEMATICAL MODEL FOR PROPOSAL SOLUTION APPROACH

The problem statement involves assigning needed tasks and workers that must be not violating an imposed constraint into a given workstation with respect to the main objective of minimizing cycle time, to improve the efficiency of throughput of a line in the real-world. For this purpose, the extension adopted a novel proposal solution approach which represents balancing mixed-model assembly line in solving worker-tasks stochastic assigned into given workstations (W – TSAGW) model. This approach is organized over three stages. In the first stage, estimating stochastic processing time using fuzzy logic, while in the second one the individual precedence graph is converted into a combined graph then so generate sequence vector (SV), third stage devoted to developing the recursive heuristic algorithm. Further details are described in the next subsections and can be summarized in Figure (1).

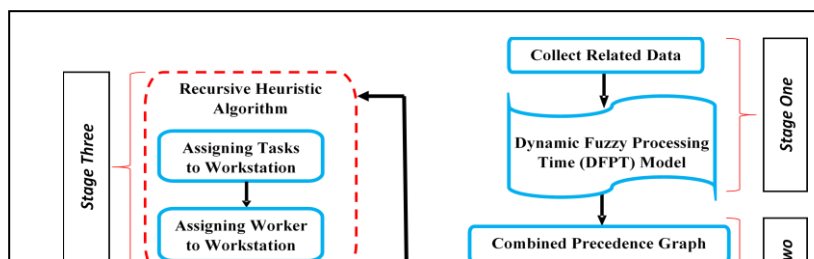


Figure 1: Framework of the W-TAWH model

4.1 Stage one:

To handle stochastic mechanism in process uncertainty, ambiguity, and variability of processing time in the real production system, this stage aimed at responding to the need by introducing a fuzzy model represented skill level (SL) and work stability(WS) variables, according to author's knowledge they are not mentioned in literature. These variables are characterized and subjected to the nature of human intuition. Thus, this study developed a dynamic fuzzy processing time (DFPT) model, the basic idea is to treat the related input data employing fuzzy logic theory. The dynamic fuzzy processing time (DFPT) model is designed to treat the difference in the max value of processing time (PT), when tuning a max value in each fuzzy set as a dynamic process. SL & WS variables are used as input variables for DFPT model. Input/output variables are fuzzified into a set of triangular and trapezoidal membership function shape in the range [0-1], whilst the output is representing the processing time of each task (i) performed by the worker (k) (PT_{ik}) as stochastic. The output variable (PT_{ik}) is controlled using 15 fuzzy rules reasoning formulas (IF <condition> THEN <result>). Finally, the Mamdani interference method was employed to fuzzy logic of DFPT model to get crisp value (PT_{ik}), see Figure (2). MATLAB SOFTWARE used for coding examined the problem.

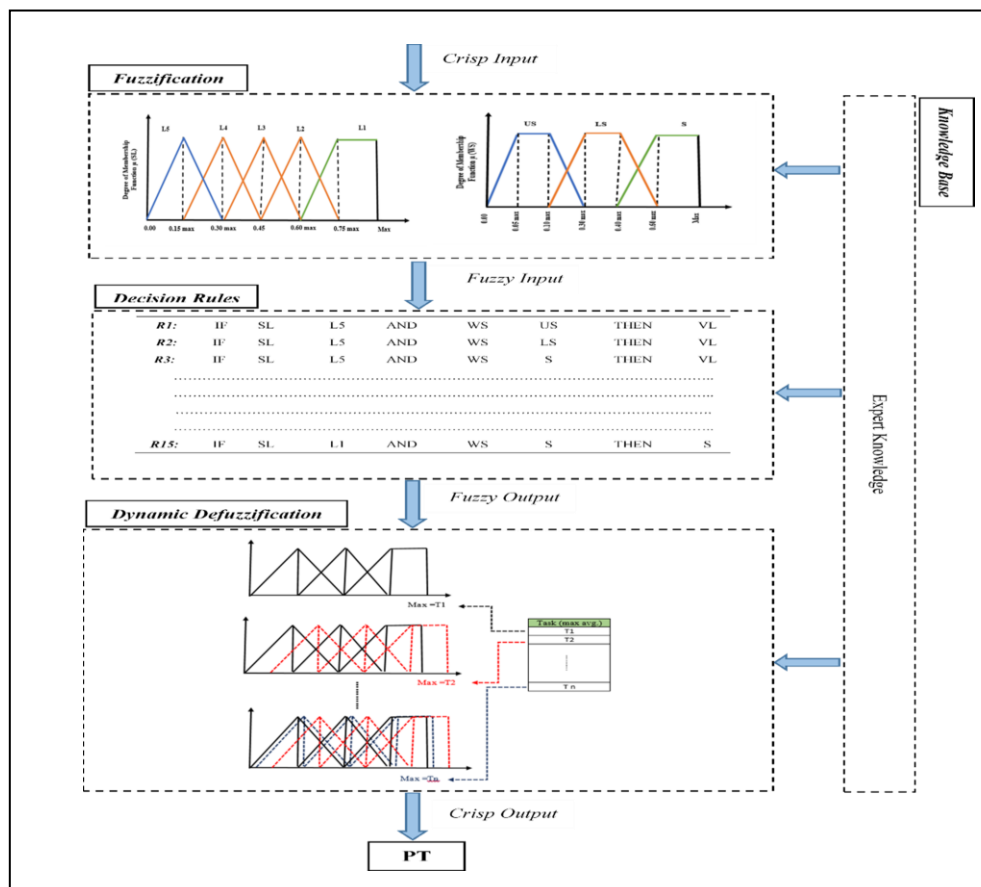


Figure 2: Outline of the developed DFPT model

4.2 Stage two

This stage is associated to collect and filter the required data that combining mix products through creating a combining matrix for computing the total time of task (TT_{ik}) using equation (1), and then computing the average process time of task (APT_i) using equation (2). Before assigning a process of a mixed assembly line, a preparation procedure must be occur; it involves a combined precedence graph to form a mix graph then to form a sequence vector (SV) using heuristic priority rules for ensuring precedence relations that imposed to find a feasible solution. Heuristic priority rules are used to rank tasks in the form SV according to their priority function and precedence among them. Tables I and II depicted the priority rules and their priority function for the S-ALBP model, and for U-ALB model.

$$TT_{ik} = \sum_{j=1}^P \sum_{k=1}^W \sum_{i=1}^n t_{ikj} \tag{1}$$

$$APT_i = \sum_{i=1}^n \sum_{K=1}^W TT_{ik} / W \tag{2}$$

TABLE I: Heuristic priority rules for S-ALB model

Heuristic Rules	Priority Function
Maximum total number of successor tasks	$P_{ts(max)} = \max \sum_{i \in S} i$
Minimum total number of predecessor task	$P_{tp(min)} = \min \sum_{i \in p} i$

TABLE II: Heuristic priority rules for U-ALB model

Heuristic Rules	Priority Function
Max total number of successor or predecessor tasks	$P_{maxf(c)} = \max \{ \text{number of task} \in \mu_c^s, \text{number of tasks} \in \mu_c^p \}$
Min total number of successor or predecessor tasks	$P_{minf(c)} = \min \{ \text{number of tasks} \in \mu_c^s, \text{number of tasks} \in \mu_c^p \}$

4.3 Stage three

This stage aims to find the primary objective of a workable solution regarding achieving the minimum cycle time for assigning task - worker to given workstations along the assembly line. To attain the main objective of the problem is conducted by adopting recursive heuristic algorithm, searching process based on maximum equality in distributing total processing time across fixed number workstations, so to achieve maximum equality time between workstations through partitioning SV data. Hence slight variation among workstations times will occur. For more clarification, the stepwise procedure of the developed algorithm is detailed below:

4.3.1 Assigning tasks to workstations

Assigning tasks into workstations of the developed algorithm is detailed below and enhanced by the Figure (3).

Step-1: Divide the SV into A&B parts by dividing the number of workstations by 2.

Step-2: Specify the workstation ratio (WR) showing the ratio of the number of workstations assigned to each part that subject to impose a condition, i.e., $WR \leq 1$.

Step-3: Calculate time ratio (TR), display the data ratio allocated for each part of the A & B, and the definition is based on splitting the SV into two cross-vectors and can be represented by left position (PL) and the right position (PR), as shown in equation (3).

$$TR = \sum_{j=PL}^i APT_j / \sum_{j=i+1}^{Pr} APT_j \tag{3}$$

Step-4: Checking the condition that says (TR average WR) if valid, a new position (i+1) must be added and going to step (6) if not, to ensure the time allocated to each cross-vectors to minimize variation.

Step-5: The last position (i) should be omitted from the cross-vector (A), to make it not violating the $TR \leq WR$ condition.

Step-6: From 1-6 steps were repeated until the rest of the workstations become 1, that is to say each cross-vector refers to the workstation filled with a set of tasks.

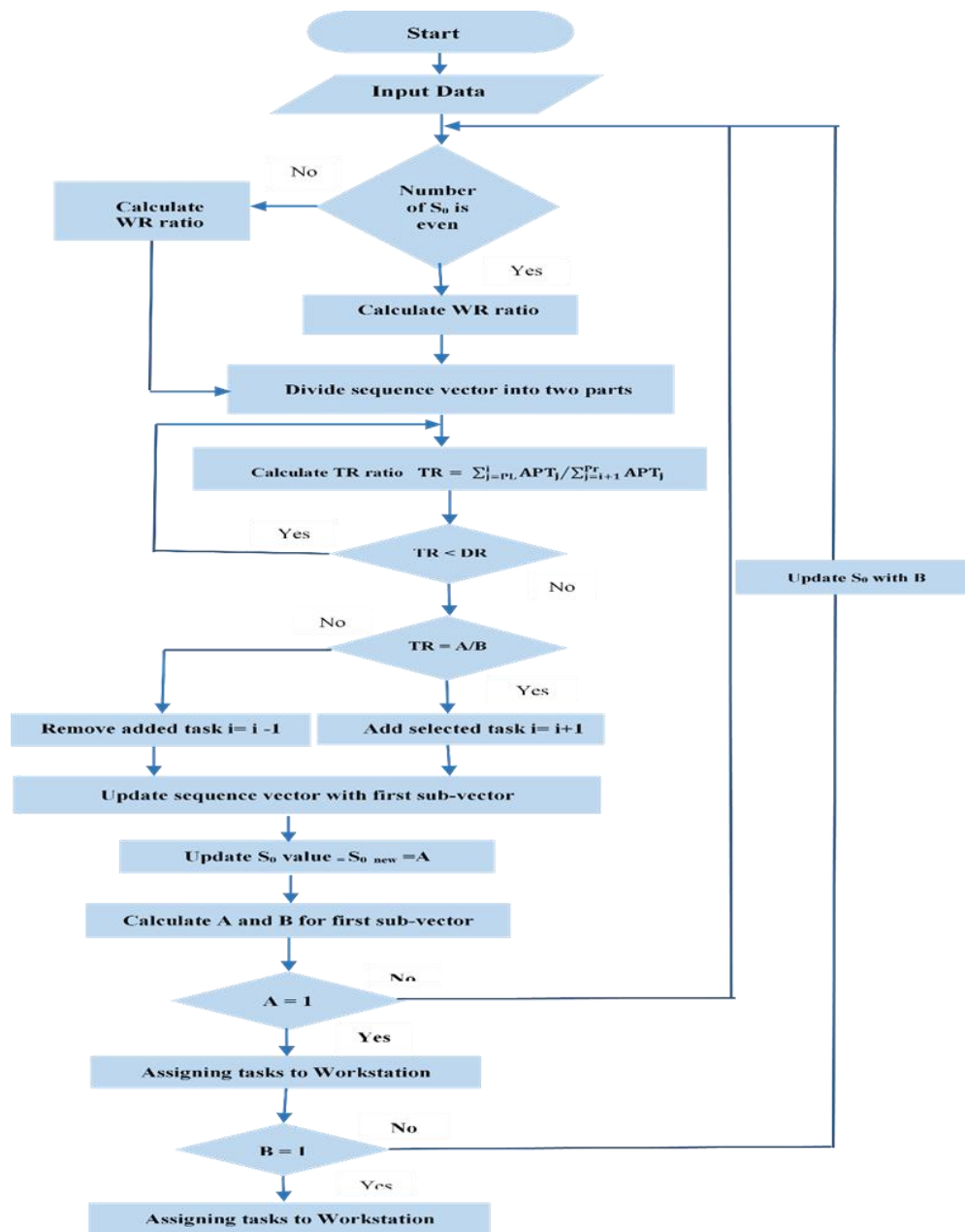


Figure 3: Flowchart of the recursive algorithm for assigning tasks to workstations

4.3.2 Assigning worker to workstation and evaluation

The mechanism of this assignment aims to minimize the cycle time for balancing the problems associated with workers assigned to given workstations and summarized as follows with help of Figure (4).

Step-1: Evaluation of workstation time ($T_s(w)$), as developed in equation (4), shows the total time required to perform the tasks assigned to the workstation.

$$T_{s(w)} = \sum_{i=1}^n \sum_{i \in S} TT_{ki} A_{Si} \quad \text{for } k = 1, \dots, W \quad (4)$$

Step-2: Invoke step (1) for all workers until all usable workers are already allocated to a workstation according to minimum $T_s(w)$.

Step 3: For all workstations repeated the steps above.

Step 4: In the final, the minimum cycle time of the assembly line is calculated through equation (5).

$$CT = \max (T_{sw}) \quad \text{for } \forall s \in S \quad (5)$$

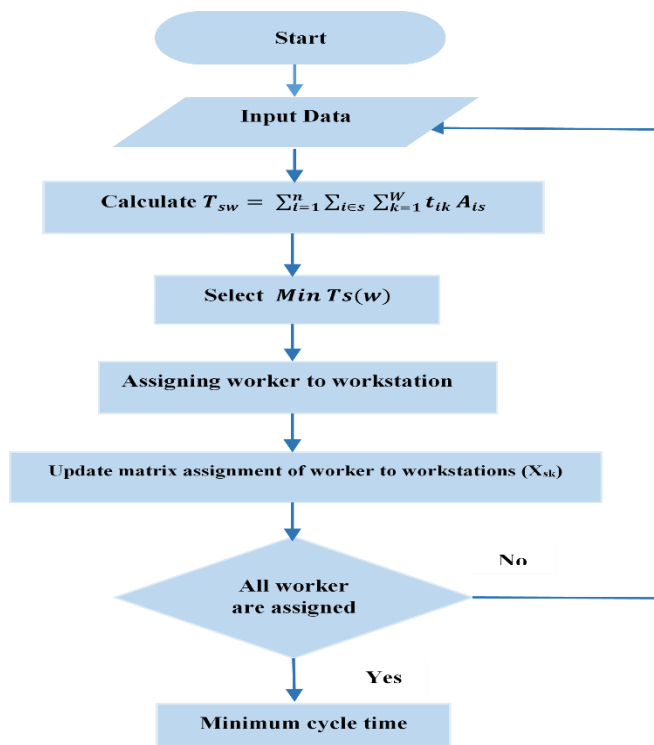


Figure 4: Flowchart of the recursive algorithm for assigning workers to workstations

5. NUMERICAL ILLUSTRATION

To demonstrate the applicability and effective solution of the proposed approach showed by a numerical example and the computational study was conducted over a set of straight and U-shaped assembly line balancing problems which are adopted from open literature. All necessary data of the illustrative example was covered in the research of Aufy and Kassam (2020) [17]. Figure (5) clarify the precedence relations among tasks of the tested problem, which had 10 tasks the nodes show the task, and the arrow represents the direction of the process. Tables III and IV show processing time for each task with different capabilities and can be performed by any one of four workers are considered. In this section, the proposed approach has been evaluated on two classes (class A & class B) to cover the diversity of the work environment are assumed for both studied layouts of mixed assembly line balancing problem. These classes were solved concerning numerical values specified with ranges represented universe of discourse the above mentioned of fuzzy sets and the final decision is focusing on estimated processing time.

Class A: Constant Skill Level (SL), Variable Work Stability (WS)

Data for three test problems (A1, A2, A3) are examined. Each test will benchmark the performance of the proposed approach through the three examined WS levels (US, LS, S) as follows:

- i. A1 \Rightarrow SL is L5 & WS is US & LS & S
- ii. A2 \Rightarrow SL is L3 & WS is US & LS & S
- iii. A3 \Rightarrow SL is L1 & WS is US & LS & S

Class B: Constant Work Stability (WS), Variable Skill Level (SL)

This case examined data for three test problems (B1, B2, B3), each test will benchmark the performance of the proposed approach through the three examined SL levels (L5, L3, L1) as follows:

- i. B1 \Rightarrow WS is US & SL is L5 & L3 & L1

- ii. B2 ➡ WS is LS & SL is L5 & L3 & L1
- iii. B3 ➡ WS is S & SL is L5 & L3 & L

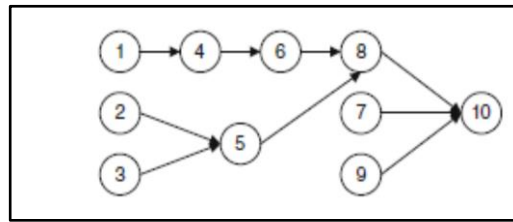


Figure 5: Combined diagram of the mix product

TABLE III: Data set of product (A)

Task No.	Task Processing Time			
	W1	W2	W3	W4
1	17	23	17	13
2	0	0	0	0
3	22	15	27	25
4	0	0	0	0
5	21	25	16	32
6	28	18	20	21
7	42	28	23	34
8	17	23	40	25
9	19	18	17	34
10	16	27	35	26

TABLE IV: Data set of product (B)

Task No.	Task Processing Time			
	W1	W2	W3	W4
1	18	22	19	13
2	21	22	16	20
3	12	25	17	15
4	29	21	19	16
5	31	25	26	22
6	0	0	0	0
7	0	0	0	0
8	27	33	40	25
9	19	13	17	34
10	26	27	35	16

The crisp value of each fuzzy set belonging to only the input variable under study denoted in the Table V, while its subjected to dynamic process in case output variable as detailed above. Then, the maximum values of fuzzy input & output variables are determined and would be listed in the Table VI. The fuzzy inference will be coded using MATLAB language.

TABLE V: Fuzzy sets ranges of the DFPT model

Fuzzy Set	Skill Level (SL)		Fuzzy Set	Work Stability (WS)	
	Boundary	Values		Boundary	Values
L5	0.00 × 0.95	0	US	0.00 × 0.8	0
	0.15 × 0.95	0.142		0.05 × 0.8	0.04
	0.30 × 0.95	0.285		0.10 × 0.8	0.08
	/	/		0.30 × 0.8	0.24
L3	0.30 × 0.95	0.285	LS	0.10 × 0.8	0.08
	0.45 × 0.95	0.427		0.30 × 0.8	0.24
	0.60 × 0.95	0.570		0.40 × 0.8	0.32
	/	/		0.6 × 0.8	0.48
L1	0.30 × 0.95	0.570	S	0.40 × 0.8	0.32
	0.75 × 0.95	0.712		0.6 × 0.8	0.48
	1.00 × 0.95	0.95		1.00 × 0.8	0.80
	/	/		1.00 × 0.8	0.80

TABLE VI: Data for setting input/output fuzzy variables

6.

Description	Data
Maximum Skill Level Value	0.95
Maximum Work Stability Value	0.80
Maximum Processing Time	Dynamic Process

COMPUTATIONAL RESULTS AND DISCUSSION

To validate the performance of the proposed approach, this section devoted to a numerical illustration has been solved. Firstly, the examined problem was ordered by two heuristic priority rules as listed above. The results obtained represent tested problems discussed over two examined layouts under study, all data tested outcome 24 alternative design solutions to solve the related problem; Tables VII and VIII summarized the results of S-ALB and U-ALB are obtained by the proposed approach, the first column shows given workstations. The second and third columns denoted the assigned workers and tasks respectively. The fourth column gives the minimum cycle time.

To assist decision making in finding the best solution in filed ALBP, Therefore, this approach study the most important assignment variables and their effect that will contribute in case balancing assembly line. These variables can be summarized are as follows: skill level, work stability, type layout, and heuristic priority rule. All results obtained benchmarked and analyzed with respects to objective function (minimum cycle time) and then categories based on studied variables are as follows:

- a. About 100% of tested problems (case A & case B) prove superiority when they take higher SL scores while WS take the lower or higher score for the examined fuzzy sets, other words SL was the more impact due to its relationship with manual and semi-automated assembly line.
- b. About 100% of tested problems (case A & case B) for U-shaped layout showed higher performance in comparison with other ones with respect to minimize cycle time.
- c. Further investigation associated with the effect of using heuristic priority rules,
 1. The maximum total number of predecessor tasks priority rule and maximum total number of successor tasks priority rule were yielded a different effect about 50 % between tested priority rules in S-ALB layout.
 2. Whilst the max total number of successor or predecessor tasks priority rule result was yielded outperform about 83% overmax number of immediate successor or immediate predecessor tasks priority rule which participates proportion about 33% in U-ALB layout.

TABLE VII: The obtained results of maximum total number of predecessor tasks priority rule and max total number of successor or predecessor tasks priority rule

Straight				U - Shaped			
Test Problem: A1				Test Problem: A1			
Workstation	Assigned Worker	Assigned Tasks	CT	Workstation	Assigned Worker	Assigned Tasks	CT
1	1	10 8 4 7	47.3	1	1	10 1 5 9	42.2
2	3	0 6 9 3		2	3	0 2 6 8	
3	4	0 5 0 2		3	4	0 3 7 0	
4	2	0 0 0 1		4	2	0 4 0 0	
Test Problem: A2				Test Problem: A2			
1	1	10 6 4 7	36.8	1	2	10 2 5 7	31.6
2	4	8 5 9 3		2	1	1 3 6 9	
3	3	0 0 0 2		3	4	0 4 0 8	
4	2	0 0 0 1		4	3	0 0 0 0	
Test Problem: A3				Test Problem: A3			
1	4	10 8 4 7	21.1	1	3	10 1 4 9	13.7
2	2	0 6 9 3		2	2	0 2 5 8	
3	4	0 5 0 2		3	4	0 3 6 0	
4	1	0 0 0 1		4	1	0 0 7 0	
Test Problem: B1				Test Problem: B1			
1	3	10 8 4 3	59.4	1	1	10 1 5 9	51.8
2	1	0 6 9 2		2	4	0 2 6 8	
3	4	0 5 7 1		3	3	0 3 7 0	
4	2	0 0 0 0		4	2	0 4 0 0	
Test Problem: B2				Test Problem: B2			
1	2	10 8 4 7	57.1	1	4	10 2 4 9	49.3
2	3	0 6 9 3		2	2	1 3 5 8	

3	4	0 5 0 2		3	3	0 0 6 0	
4	1	0 0 0 1		4	1	0 0 7 0	
Test Problem: B3				Test Problem: B3			
1	2	10 8 4 3	56.5	1	4	10 1 5 9	47.9
2	4	0 6 9 2		2	2	0 2 6 8	
3	3	0 5 7 1		3	3	0 3 7 0	
4	1	0 0 0 0		4	1	0 4 0 0	

TABLE VIII: The obtained results maximum total number of successor tasks priority rule and max number of immediate successor or immediate predecessor tasks priority rule

Straight				U - Shaped			
Test Problem: A1				Test Problem: A1			
Workstation	Assigned Worker	Assigned Tasks	CT	Workstation	Assigned Worker	Assigned Tasks	CT
1	1	1 4 7 9	40.7	1	1	10 9 5 4	34.5
2	3	2 5 8 10		2	3	0 8 7 3	
3	4	3 6 0 0		3	4	0 0 6 2	
4	2	0 0 0 0		4	2	0 0 0 1	
Test Problem: A2				Test Problem: A2			
1	2	1 3 6 9	38.6	1	1	10 9 7 2	31.6
2	3	2 4 7 10		2	2	0 8 6 1	
3	1	0 5 8 0		3	4	0 5 4 0	
4	4	0 0 0 0		4	3	0 0 3 0	
Test Problem: A3				Test Problem: A3			
1	3	1 5 7 9	13.7	1	3	10 5 6 2	12.6
2	2	2 6 8 10		2	4	9 7 4 1	
3	4	3 0 0 0		3	2	8 0 3 0	
4	1	4 0 0 0		4	1	0 0 0 0	
Test Problem: B1				Test Problem: B1			
1	1	1 4 6 8	60.2	1	4	10 9 5 4	55.1
2	2	2 5 7 9		2	2	0 8 7 3	
3	4	3 5 7 9		3	1	0 0 6 2	
4	3	0 0 0 10		4	3	0 0 0 1	
Test Problem: B2				Test Problem: B2			
1	2	1 3 6 9	55.1	1	1	10 8 7 4	52.6
2	3	2 4 7 10		2	2	9 5 6 3	
3	4	0 5 8 0		3	3	0 0 0 2	
4	2	0 0 0 0		4	4	0 0 0 1	
Test Problem: B3				Test Problem: B3			
1	1	1 5 7 9	53.3	1	4	10 5 6 2	44.7
2	3	2 6 8 10		2	3	9 7 4 1	
3	4	3 0 0 0		3	2	8 0 3 0	
4	2	4 0 0 0		4	1	0 0 0 0	

7. CONCLUSIONS

Modeling the ALBP has covered a wide range of real-world industries. The recent advances have created opportunities to solve a more challenging problem. In this study, we have formulated a mixed model of straight and U-shaped assembly line balancing in a stochastic work environment. Processing time for assigned tasks consistent with the accumulated experience of workers presented in the assumption that says each worker can perform the assembly task with a different processing time. Recursive heuristic algorithms were adopted in solving ALBP based on dividing data that aiming to minimize cycle time with the secondary objective is reducing the number of workstations and workers. Finally, the proposed approach was examined in a theoretical application taken from the literature. The primary results were proof of its validity in studying and identifying the important role of the stated assembly line balance considerations and indicated that the solutions make the algorithm

get perfect performance whenever the specific type of assembly line considerations in comparing with tested others.

Acknowledgment

The authors would like to thank the University of Technology particularly Production Engineering and Metallurgy Department for facilities in preparing the paper.

References

- [1] M. H. Alavidoost, H. Babazadeh, and S. T. Sayyari, "An interactive fuzzy programming approach for bi-objective straight and u-shaped assembly line balancing problem", *Applied Soft Computing*, Vol. 40, Issue 3, pp. 221-235, March 2016.
- [2] M. H. Alavidoost, and M. H. F. Zarandi, M. Tarimoradi, and Y. Nemati, "Modified genetic algorithm for simple straight and u-shaped assembly line balancing with fuzzy processing times", *Journal of Intelligent Manufacturing*, Vol. 28, Issue 7, pp. 313-336, 2017.
- [3] T. Pape, "Heuristics and lower bounds for the simple assembly line balancing problem type 1: overview computational tests and improvements", *European Journal of Operational Research*, Vol. 240, No. 1, pp. 32-42, 2014.
- [4] U. Özcan, and B. Toklu, "A new hybrid improvement heuristic approach to simple straight and u-type assembly line balancing problems", *Journal of Intelligent Manufacturing*, Vol. 20, No. 123, pp. 123-136, 2009.
- [5] E.P.C. Kao, "A preference order dynamic program for stochastic assembly line balancing", *Journal of Management Science*, Vol. 22, No. 10, pp. 1097-1104, 1976.
- [6] C. Becker, and A. Scholl, "A survey on problems and methods in generalized assembly line balancing", *European Journal of Operational Research*, Vol. 168, No. 8, pp. 694-715, 2006.
- [7] A. Scholl, and C. Becker, "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing", *European Journal of Operational Research*, Vol. 168, Issue 3, pp. 666-693, 2006.
- [8] Y. Monden, "Toyota production system: an integrated approach to just-in-time", CRC Press Taylor & Francis Group, Fourth Edition, New York, 2011.
- [9] Y. Monden, "Toyota production system: practical approach to production management", Industrial Engineers and Management Press, Norcross, Georgia, First Edition, 1983.
- [10] P. M. D. S. A Pedro Costa., "A genetic algorithm approach for the type II assembly line balancing problem", M. Sc. Thesis, Mechanical Engineering, TÉNICO Lisboa University, Lisboa, Spain, June 2016.
- [11] P.Th. Zacharia, and A. C. Nearchou, "A meta-heuristic algorithm for the fuzzy assembly line balancing type-E problem", *Computer and Operation Research*, Vol. 40, Issue 12, pp. 3033-3044, 2013.
- [12] A. Attia, "Solving assembly line balancing problems using genetic algorithm", M.Sc. Thesis, Mechanical Production and Design, Benha University, Benha, Egypt, June 2006.
- [13] Y. Kara, T. Paksoy, and C-T. Chang, "Binary fuzzy goal programming approach to single model straight and U-shaped assembly line balancing", *European Journal of Production Research*, Vol. 195, Issue 2, pp. 335-347, 1 June 2009.
- [14] S. Avikal, R. Jain, P. K. Mishra, and H. C. Yadav, "A heuristic approach for U-shaped assembly line balancing to improve labor productivity", *Computers & Industrial Engineering*, Vol. 64, Issue, pp. 895-901, April 2013.
- [15] H. P. Toroudi, M. S. Madani, F. Sarlak, and Y. G. Kanani, "A multi-objective method for solving assembly line balancing problem", *Decision Science Letters*, Vol. 6, Issue 1, pp. 1-10, 2017.
- [16] D. Krenczyk and K. Dziki, "A hybrid heuristic algorithm for multi-manned assembly line balancing problem with location constraints", 14 International Conference on Soft Computing Models in Industrial and Environmental Applications, Seville, Spain, pp. 333-343, May 13-15, 2019.
- [17] S. A. Aufy, and A. H. Kassam, "A consecutive heuristic algorithm for balancing a mixed-model assembly line type II using a (W-TAWH) model developed for straight and U-shaped layouts", *Material Science and Engineering*, Vol. 671, January 2020, DOI: 10.1088/1757-899X/671/1/012147. [Online]. Available: <https://iopscience.iop.org>.