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

Samah A. Aufy and AllaEldin H. Kassam

Abstract. The development of assembly lines in manufacturing systems has evolved due to consumer expectations and a growing and increasingly competitive global market. The assembly line balancing problem (ALBP) is well recognised in manufacturing and may arise when tasks are assigned to the workstation. In this paper, a problem statement is structured by assigning a task to a worker in a suitable workstation, subject to some constraints, in order to reduce the cycle time and improve performance. This is a new approach, based on worker and task assignment and is thus a worker–task assigned to workstation heuristic model (W-TAWH). It has been developed for straight and U-shaped models. The methodology is described as two stages, the input related data stage and assignment stage. Firstly, the focus is on combining precedence diagram and generating the sequence vector procedure. The second stage comprises development of comprehensive mathematical formulae for the consecutive algorithm to address the straight and U–shaped balancing problem, with minimization or maximization of relevant criteria. This generates performance evaluation criteria according to the number of tasks and workers assigned to the workstation, and these criteria are integrated into a single score using a desirability function approach. Finally, the performance of the developed algorithm is validated through a numerical example. We tested 24 alternative solutions and the fourth solution (1-4-1) achieved the highest score (0.726), which represented the following assignment variables: straight layout, minimum total number of successor tasks rule, and four workers. The performance evaluation is represented by the lower bound of cycle time and upper bound of efficiency.

Keywords: Heuristic algorithm, Mixed-model assembly line, Cycle time, Worker, Desirability function

1. Introduction
The development of assembly lines in manufacturing systems has changed over time, in response to consumer expectations and an increasingly competitive global market. This compels manufacturing organizations to explore productivity improvement strategies, which include assembly line balancing problems (ALBP) [1] [2].

The simple assembly line balancing problem considers the process of assembling a product as a sequence of tasks that must be assigned to workstations. The assignment of tasks is based on the time units applicable to finishing the task, and the precedence relationships among them.
The ALBP generates two types of challenge. The first of these is the type I problem, in which the optimization objective is to minimize the number of workstations with respect to a given cycle time. This is particularly relevant to the design of new assembly lines. The second problem is type II, where the objective lies in minimizing the cycle time for a given number of workstations. This is more relevant to rebalancing an existing assembly line.

A further classification is based on the assembly line layout. Two of the types available are (a) the straight line, whereby tasks are assigned only where predecessors are already assigned to the workstation and (b) U-shaped, whereby tasks are assigned where predecessors and successors are already assigned to a workstation. It can be inferred that efficiency of a solution for a U-shape should be better for the same number of workstations, because there are more options of assignable tasks available than for a straight assembly line [3] [4]. Figure 1 shows types of assembly line based on layout.

Due to the need to satisfy customers with a variety of products delivered in a timely manner, and rapidly-changing markets, the assembly line has tended towards a mixed model because this allows the production of different models on the same line, in small batches [1]. The mixed-model assembly line balancing (MMALB) problem can be described as: given P products, a set of tasks associated with each product, task time, and also sets of precedence relationships among tasks for each one. General purpose machines with automated tool changes and highly flexible workers are necessary, to generate a mixed sequence whereby various models with similar process requirements are assembled on the same line [5].

Recently, mixed-model and various layouts have been tried in a bid to solve the assembly line balancing problem. Furthermore, classification of workers according to skills and experiences based on qualifications such as employment period and training period have been used in the assignment tasks to the workstation. The questions of how best to assign suitable workers and tasks to workstations in such a way that these improve the efficiency of the assembly line and decrease the cycle time by reducing task time are complex challenges within the assembly line balancing problem.

![Figure 1. Task assignment in types of assembly line: (a) straight assembly line (b) U-shaped assembly line [1]](Image)

The problem of designing a mixed model, and the assembly line balancing problem, have been extensively examined in the literature. Among numerous relevant, published studies are Avikal et al. (2013) [6]. These authors introduced an analysis of labour productivity for both U-shaped and straight lines, carried out via a proposed heuristic based on the critical path method for the assignment of tasks to the workstations on a U-shaped assembly line. The results of this heuristic are a reduced minimum number of workstations in the U-shaped line as compared to the straight line, and an improved labour productivity level.

Zhang and Cheng (2015) [1] proposed a heuristic procedure to solve the mixed model U-line balancing problem (MMULBP), with processing time and cycle time as fuzzy numbers. This procedure uses a
ranked positional weight rule and some improvement in order to cover a two-sided search for U-shaped layout. The yield of the experimental example proved that the proposed procedure is more effective.

Mamun et al. (2012) [7] used the meta-heuristic genetic algorithm to produce a procedure for balancing mixed model assembly line type I with production rates and some common features, in particular parallel workstation, resource limitation and zoning constraints.

Toroudi et al. (2017) [8] formulated a mixed integer non-linear optimization to solve multi-objective decision-making problems with cost and cycle time for simple assembly line balancing. Hybrid algorithms, based on practical swarm optimization and simulated annealing, were used. The results of the multi-objective particle swarm optimization provided the best quality solution, while the primary result of hybrid method gave a more non-dominated Pareto solution.

However, previous researchers have not considered the changes in task time associated with worker skill and experience of ALBP. The purpose of this paper is to solve the problem described above, and the paper is structured as follows: in the first section, a mathematical model of the worker–task assigned to the workstation heuristic is formulated. Then, a desirability function approach is adopted, to integrate the performance measures into one score. Next, the numerical example is used to validate the performance of the developed algorithm.

2. Mathematical Formulation of Problem Statement
In this paper, a problem statement can be structured by assigning a proper task to an worker and, subject to some constraints, determining a suitable constant workstation number in order to reduce the cycle time; in other words, to increase the efficiency of total throughput. The extension of a mixed model assembly line balancing problem type II (W-TAWH) is thus developed with respect to straight and U-shaped models.

In general, the outline solution to the W-TAWH problem may be described using two stages: an input-related data stage and an assignment stage. The first stage is required to combine the precedence diagram procedure and to generate sequence vector procedure. The assignment stage is to develop a comprehensive mathematical formula for a consecutive algorithm for assigning process (tasks and worker) with the objective to minimize the cycle time of given workstations for both straight and U-shaped models, as shown in Figure 2.

Four assumptions used in the W-TAWH model must be considered:

1. The sequence of tasks is essential; therefore, their precedence constraints meet among them.
2. A task assigned to only one workstation.
3. Assignation of a single worker to each workstation. Thus, the number of workstations is equal to the number of workers and that maximises resource use.
4. Task time differs among workers, because of differences in their work experience and capabilities.
2.1. Stage One: Input-Related Data

The input-related data stage is essential for combining mixed models into a precedence diagram for combined and mixed different product models, and also for generating sequence vector (sv), before assigning task–worker for a mixed assembly line model. To fix the number of workstations a suitable procedure must be created, to form the sv to handle the precedence relationships that are imposed, and to find a feasible solution.

In the literature, a list of heuristic priority rules is used to rank the set of tasks in the form sv, according to their priority function and the precedence relationships among them. Table 1 shows the priority rules for a straight assembly line, while Table 2 shows the priority rules for a U-shaped assembly line.

Table 1. Set of heuristic priority rules for the straight assembly line model

| Heuristic Rules                      | Priority Function                                      |
|-------------------------------------|--------------------------------------------------------|
| Maximum task time                   | $p_t^{\text{max}} = \max t_i$                         |
| Minimum task time                   | $p_t^{\text{min}} = \min t_i$                         |
| Maximum total number of successor tasks | $p_{st}^{\text{max}} = \max \sum_{i \in s} t_i$  |
| Minimum total number of successor tasks | $p_{st}^{\text{min}} = \min \sum_{i \in s} t_i$  |
| Maximum total number of predecessor tasks | $p_{tp}^{\text{max}} = \max \{\sum_{i \in p} \}$ |
| Minimum total number of predecessor tasks | $p_{tp}^{\text{min}} = \min \{\sum_{i \in p} \}$ |

Table 2. Set of heuristic priority rules for a U-shaped assembly line model

| Heuristic Rules                          | Priority Function                                      |
|-----------------------------------------|--------------------------------------------------------|
| Maximum task time                       | $p_{\text{max}}(c) = t(c)$                           |
| Minimum task time                       | $p_{\text{min}}(c) = t(c)$                           |
| Max number of immediate successor or immediate predecessor tasks | $p_{\text{maxIF}}(c) = \max (\mu_{cs}^{\text{IF}}, \mu_{cp}^{\text{IF}})$ |
| Min number of immediate successor or immediate predecessor tasks | $p_{\text{minIF}}(c) = \min (\mu_{cs}^{\text{IF}}, \mu_{cp}^{\text{IF}})$ |
| Max total number of successor or predecessor tasks | $p_{\text{max}}(c) = \max \{\text{number of task } \in \mu_{cs}^{\text{IF}} \}$ \mid \text{number of tasks } \in \mu_{cp}^{\text{IF}} \}$ |
Min total number of successor or predecessor tasks  \[ P_{\text{min}}(c) = \min \left\{ \text{number of tasks } \in \mu_c^p, \text{number of tasks } \in \mu_c^s \right\} \]

Where \( \mu_c^p \) set of tasks (p) that precede task c, \( \mu_c^s \) set of tasks (s) which succeed task c. Then assignable tasks = \{c\} all \( i \in \mu_c^p \) or all \( j \in \mu_c^s \} \) have already been assigned.

2.2. Stage Two: Assignment
The aim of this stage is to find a feasible solution with minimal associated cycle time for the task/worker assembly line balancing problem. The minimized cycle time of W-TAWH goal is obtained by heuristic consecutive algorithm attempts to allocate tasks to a workstation, using workers that allow maximization of the number of tasks performed at each workstation by one worker, taking into considered load variation. Achievement of this fulfills the objective of this stage.

The aim of this heuristic algorithm is to assign tasks and workers to the given number of workstations, determining the minimum cycle time of the assembly line as much as possible, so that a maximal number of tasks will be assigned to the workstation within minimum cycle time (CT). The stepwise procedure of the developed consecutive algorithm is given below, and shown in Figure 3.

Step 1: Compute initial cycle time (CT_i), which can be defined as an average of the minimum execution time of worker for all products could operate the tasks.
\[ CT_i = \sum_{k=1}^{W} \min_{1 \leq k \leq W} TT_{ik} / W \]

Step 2: Compute workstation time (Ts(w)) for all workers available by summing times of maximal tasks number that performed by available workers must be within or less than the value of the initial cycle time.
\[ T_s(w) = \sum_{c=P_1}^{P_{m\text{[w]}}} TT_{ik} < CT_i \leq \sum_{c=P_1}^{P_{m\text{[w]}}} 1 TT_{ik} \]

Where, \( T_s(w) \) is a workstation time for worker (k)( k=1, ..., W), \( P_1 \) is the first position for assigned tasks to workstation, and \( m(k) \) defines the maximal number of tasks a worker (k) could operate in the given sequence during a time less than CT_i.

Step 3: The set of preferred workers \( P_{m\text{[w]}} \) for each workstation (s) (where 1≤s≤S) is defined for all workers can execute the assigning tasks within value of CT_i to satisfy the condition (\( w_k \in P_{m\text{[w]}} \) if \( \mu(c) \geq m(k) \), where 1 ≤ k ≤ W).
Where, \( w_s \) is worker selected, \( P_{m\text{[w]}} \) is the set of available workers could perform maximal number of tasks in the given sequence during a time less than CT_i, and \( m(c) \) is the maximal number of choose tasks (c) could operate by worker (k) in the set \( P_{m\text{[w]}} \).

Step 4: For workers with the same number of tasks, choose one that minimizes the total execution time of the workstation as (\( w(s) = k \), if \( T_s(k) \leq T_s(w) \) ∀ k ∈ w).
Where, \( w(s) \) is the worker to be assigned to the workstation (s), and \( T_s(k) \) is the time of workstation (s) for worker (k).

Step 5: Check if all workstations are full then continue, otherwise a new workstation is open and the start position for the next workstation is calculated as equation (3), then go to (steps 2–5) to assign remaining tasks in given sequence.
\[ P_{1s+1} = PL_{ps} + 1 \]

Let \( P_{1s+1} \) is the start position for the next workstation, and \( PL_{ps} \) is the last position of that precedence workstation.

Step 6: Check if all tasks in given sequence are assigned and if so continue, otherwise increase unit step of CT.
Step 7: Find the minimum cycle time (CT), this is the maximum total time of tasks assigned to workstation, and expressed in equation (4).

\[ CT = \max (Ts(s)) \quad \text{for} \quad \forall s \in S \quad (4) \]

Step 8: Evaluate the quality of solution means combination of the performance measures of the solution must be done are:

1. Line efficiency (E) shows the percentage use of the assembly line, and can be expressed as equation (5).

\[ E = \frac{\sum_{p=1}^{P} n_p \sum_{s=1}^{S} Ts_s}{N \times S \times CT} \times 100 \quad (5) \]

Where

- \( n_p \): number of products for product \( p \)
- \( N \): total number of products for mix products on assembly line

2. Smoothed workstation index (SI) is a fluctuation of assembly times required by each workstation per shift time, in principle, zero value of SI is an indication of perfect balance assembly line. It can be represented as in equation (6) [9].

\[ SI = \sum_{s=1}^{S} |As - Ts| \quad (6) \]

Where

- \( As \): average workstation time during shift time

3. Variation (V) is a measure of utilization between workstations. It is based on utilization variability and product variability. The ‘utilization variability’ measures the difference between utilization workstation time and average of utilization over all workstations for product \( p \). The ‘product variability’ is a measure of weighted sum of the variability of each product \( p \), it can be represented by equation (7) [10].

\[ V = \sum_{p=1}^{P} \frac{1}{N_p} \sqrt{\frac{1}{S} \sum_{s=1}^{S} (U_{sp} - \bar{U}_p)^2} \quad (7) \]

Where

- \( U_{sp} \): utilization of workstation \( s \) for product \( p \).
- \( \bar{U}_p \): average utilization of all workstations for product \( p \).

4. Idle time (Id); selecting this measure reduces the time losses during shift time, that is the differences between shift time and total assembly time required as illustrate in equations (8) [9].

\[ Id = S \times \text{shift time} - \text{TAL} \quad (8) \]

Where TAL, total assembly time required in a shift time.
3. Desirability Function Approach for Optimizing Assignment Variables

W-TAWH is conducted for a set of different assignment variables. These variables are distinguished according to the layout, priority rules and number of workers, while each variable is assigned with type and associated levels as illustrated in Table 3. Accordingly, assume that there are (h) alternative solutions for the candidate problem. The evaluation of the alternative solutions involves the simultaneous study of the performance measures such as cycle time (CT_h), efficiency (E_h), smoothness index (Sl_h), variation (V_h), and ideal time (I_d_h) measure values for each alternative solution (h=1,..., H).
Table 3. Assignment variables, types and their levels

| Assignment variable | Type                | Level   |
|---------------------|---------------------|---------|
| Layout              | Straight, U-shaped  | [1,2]   |
| Sequence Vector     | Priority Rules      | [1,2,...,6] |
| Worker Number       | Four, Three         | [1,2]   |

This desirability function approach can be used to set assignment variables to determine that the best possible combination of the solution is obtained. This approach assigns a score to set alternative solutions, and chooses the best variable set that maximizes the score [11] [12].

Generally, the desirability function approach consists of transforming individual performance measures (i) into individual desirability value (di), bonded by 0≤ di ≤ 1, depending on whether these measures (ih) are one-sided in cases maximizing or minimizing such that are smaller - the - better CTi, SIh, Vh, Ids are defined by equation (9) and those larger -the - better Eh defined by equation (10) [13]. A common approach is to combine overall individual desirability values (Dh) as the geometric mean for all individual desirability, as given by the equation (11):

\[ d_i(h) = \begin{cases} 1 & \text{if } i_h \leq d_i^{\text{min}} \\ \left(\frac{d_i^{\text{max}} - i_h}{d_i^{\text{max}} - d_i^{\text{min}}}\right)^{w_i} & \text{if } d_i^{\text{min}} < i_h < d_i^{\text{max}} \\ 0 & \text{if } i_h \geq d_i^{\text{max}} \end{cases} \]  

(9)

\[ d_i(h) = \begin{cases} 0 & \text{if } i_h \leq d_i^{\text{min}} \\ \left(\frac{i_h - d_i^{\text{min}}}{d_i^{\text{max}} - d_i^{\text{min}}}\right)^{w_i} & \text{if } d_i^{\text{min}} < i_h < d_i^{\text{max}} \\ 1 & \text{if } i_h \geq d_i^{\text{max}} \end{cases} \]  

(10)

Where

- \( d_i(h) \): the value of desirability function for performance measure (i) in solution (h).
- \( d_i^{\text{min}} \): lower bound for individual performance measure (i)
- \( d_i^{\text{max}} \): upper bound for individual performance measure (i)

\[ D_h = \left(\prod_{i=1}^{A} d_i^{m(h)}\right)^{\frac{1}{\sum_{i=1}^{A} m_i}} \]  

(11)

Where

- \( m_i \): represents the relative performance (weight) among performance measure, these weights in the research, \( m=5 \) refers to the number of individual desirability.

4. Numerical Example

All the necessary data of the example problem for distinguishing the developed approach was covered in a study by Wenqing, Zhng & Mistuuo Gen [10], with 10 tasks and two models (A & B). This study explored the assembly line balancing problem in which balancing was conducted for a mixed model and coded in the Matlab language, whereby running the program consisted of input and reading the data for relative problems. Assuming differences between workers authorized to undertake each task in terms of specific processing time according to their skill levels, as presented in Tables 4 and 5, the precedence graph of combined model is shown in Figure 4.

Table 4. Data set of model (A).

| Task No. | Task Time | W1 | W2 | W3 | W4 |
|----------|-----------|----|----|----|----|
|          |           |    |    |    |    |
Table 5. Data set of model (B).

| Task No. | Task 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        |

Figure 4. Precedence graph of combined model

5. Results and Analysis
To validate the performance of the developed W- TAWH model, a numerical example has been solved. Firstly, the test problem was ranked by six priority rules as mentioned in section 2.1, and labelled level 1, level 2, level 3, level 4, level 5 and level 6 respectively. Worker number variables treated with 4 and 3 workers were designated level 1 and level 2. Layout variables, straight and U-shaped, were labelled level 1 and level 2. These variables and their levels generated 24 alternatives in terms of the form sequence vector, as shown in Table 6. These vectors were then solved using the W-TAWH model for assigning tasks and workers to workstations.
Then, the W – TAWH model was run to solve the related problem; Table 7 shows the experimental results, note that column 1 denotes the number of workstations associated with each type of layout, columns 2 and 3 denote the ordered of assigned workers and tasks into four workstations. Table 8 shows the experimental results for the W – TAWH model, note that column 1 denotes the number of workstations associated with each type of layout, columns 2 and 3 denote the ordered of assigned workers and tasks into three workstations.

Table 6. Ranking of tasks according to six priority rules for three and four workers.

| Level of Worker Number | Levels of Layout | Levels of priority Rules | Sequence Vector |
|------------------------|------------------|--------------------------|-----------------|
| 1                      | 1                |                          | 9 3 1 7 4 6 2 5 8 10 |
| 2                      | 2                |                          | 2 7 1 4 6 3 9 5 8 10 |
| 3                      | 3                |                          | 1 2 3 4 5 6 7 8 9 10 |
| 4                      | 4                |                          | 1 0 9 8 7 6 5 4 3 2 1 |
| 5                      | 5                |                          | 1 0 8 6 5 4 9 7 3 2 1 |
| 6                      | 6                |                          | 1 2 3 7 9 4 5 6 8 10 |
| 1                      | 1                |                          | 3 1 9 7 4 6 2 5 8 10 |
| 2                      | 2                |                          | 2 7 9 1 4 6 3 5 8 10 |
| 3                      | 3                |                          | 1 2 3 4 5 6 7 8 9 10 |
| 4                      | 4                |                          | 1 0 9 8 7 6 5 4 3 2 1 |
| 5                      | 5                |                          | 1 0 8 6 5 4 9 7 3 2 1 |
| 6                      | 6                |                          | 1 2 3 7 9 4 5 6 8 10 |

| Table 7. Tasks and workers assigned to four workstations according six priority rules for straight and U-shaped assembly line. |
|-------------------------------------------------|-------------------------------------------------|
| **Straight** | **U-Shaped** |
| Rule: Max Time | Rule: Max Time |
| Workstation | Assigned Worker | Assigned Tasks | Workstation | Assigned Worker | Assigned Tasks |
| 1 | 2 | 9 3 | 1 | 1 | 10 8 |
| 2 | 4 | 1 7 4 | 2 | 3 | 5 9 |
| 3 | 3 | 6 2 5 | 3 | 4 | 3 1 |
### Table 8. Assignment of tasks and workers into three workstations according six priority rules for straight and U-shaped assembly line.

| Rule: Min Time | 4 | 1 | 8 10 | 4 | 2 | 7 6 4 2 |
|----------------|---|---|------|---|---|-------|
| Rule: Max total number of successor tasks | Rule: Min Time | 1 | 3 | 2 7 1 | 4 | 3 | 2 7 1 |
| Rule: Max number of immediate successor or immediate predecessor tasks | Rule: Min Time | 2 | 4 | 4 6 3 | 3 | 2 | 4 6 3 |
| Rule: Min total number of successor tasks | Rule: Max total number of predecessor tasks | 3 | 2 | 9 5 | 1 | 4 | 9 5 |
| Rule: Min total number of successor or predecessor tasks | Rule: Min Time | 4 | 1 | 8 10 | 4 | 1 | 10 8 |
| Rule: Max total number of predecessor tasks | Rule: Min total number of successor tasks | Rule: Max total number of successor or predecessor tasks | Rule: Min Time | 1 | 4 | 10 9 |
| Rule: Min total number of successor or predecessor tasks | Rule: Max total number of successor or predecessor tasks | 2 | 3 | 4 5 6 | 2 | 4 | 8 5 |
| Rule: Min total number of successor or predecessor tasks | Rule: Min number of immediate successor or immediate predecessor tasks | 3 | 2 | 7 8 | 3 | 3 | 7 6 4 3 |
| Rule: Min total number of successor or predecessor tasks | Rule: Min number of immediate successor or immediate predecessor tasks | 4 | 1 | 9 10 | 4 | 2 | 2 1 |
| Rule: Max total number of predecessor tasks | Rule: Max total number of successor or predecessor tasks | Rule: Min number of immediate successor or immediate predecessor tasks | Rule: Min Time | 1 | 4 | 10 2 |
| Rule: Min total number of successor or predecessor tasks | Rule: Min number of immediate successor or immediate predecessor tasks | 2 | 3 | 6 5 4 | 2 | 2 | 3 4 |
| Rule: Min total number of successor or predecessor tasks | Rule: Min number of immediate successor or immediate predecessor tasks | 3 | 2 | 9 7 | 3 | 3 | 5 6 7 |
| Rule: Min total number of successor or predecessor tasks | Rule: Min number of immediate successor or immediate predecessor tasks | 4 | 1 | 3 2 1 | 4 | 1 | 8 6 |

| Straight | U - Shaped |
|----------|------------|
| Rule: Max Time | Rule: Max Time |
| Workstation | Assigned Worker | Assigned Tasks | Workstation | Assigned Worker | Assigned Tasks |
| 1 | 1 | 3 1 | 1 | 1 | 10 8 |
| 2 | 3 | 7 4 6 2 5 | 2 | 3 | 5 3 1 |
| 3 | 2 | 8 10 | 3 | 2 | 9 7 4 6 2 |
| Rule: Min Time | Rule: Min Time |
| Workstation | Assigned Worker | Assigned Tasks | Workstation | Assigned Worker | Assigned Tasks |
| 1 | 3 | 2 7 9 1 4 | 1 | 3 | 2 7 9 1 4 |
| 2 | 2 | 6 3 5 | 2 | 2 | 6 3 5 |
| 3 | 1 | 8 10 | 3 | 1 | 8 10 |
| Rule: Max total number of successor tasks | Rule: Max number of immediate successor or immediate predecessor tasks | Rule: Min total number of successor tasks | Rule: Min number of immediate successor or immediate predecessor tasks |
| Workstation | Assigned Worker | Assigned Tasks | Workstation | Assigned Worker | Assigned Tasks |
| 1 | 3 | 1 2 3 4 | 1 | 1 | 10 9 8 |
| 2 | 2 | 5 6 7 | 2 | 3 | 5 7 6 4 |
| 3 | 1 | 8 9 10 | 3 | 2 | 3 2 1 |
| Rule: Min total number of successor tasks | Rule: Min number of immediate successor or immediate predecessor tasks | Rule: Max total number of predecessor tasks | Rule: Max total number of successor or predecessor tasks |
| Workstation | Assigned Worker | Assigned Tasks | Workstation | Assigned Worker | Assigned Tasks |
| 1 | 1 | 10 9 8 | 1 | 1 | 2 3 7 |
| 2 | 3 | 7 6 5 4 | 2 | 2 | 9 10 4 |
| 3 | 2 | 3 2 1 | 3 | 1 | 5 8 6 |
| Rule: Max total number of predecessor tasks | Rule: Max total number of successor or predecessor tasks |
To demonstrate the quality of improved T–WAWH, testing was performed on 24 solutions of mixed ALBP. The obtained results are shown in Table 9. The performance measures for five measures and 24 alternatives are considered indications of the expected quality of solutions, comparing the results of experimental study yielded by the desirability function.

### Table 9. Performance measures for all alternative solutions.

| NO. | Level | Performance Measures |
|-----|-------|----------------------|
|     |       | CT<sub>h</sub>| E<sub>h</sub>| SI<sub>h</sub>| Id<sub>h</sub>| V<sub>h</sub> |
| 1   | 1-1-1 | 43       | 0.883 | 207.39 | 532.4 | 1.22          |
| 2   | 1-2-1 | 43       | 0.919 | 186.5  | 470.9 | 1.06          |
| 3   | 1-3-1 | 43       | 0.94   | 197.64 | 372.7 | 1.184         |
| 4   | 1-4-1 | 43       | 0.944 | 197.64 | 372.7 | 1.184         |
| 5   | 1-5-1 | 43       | 0.893  | 278.03 | 512.6 | 1.572         |
| 6   | 1-6-1 | 43       | 0.903  | 210.11 | 496.3 | 1.128         |
| 7   | 1-1-2 | 60       | 0.924  | 101.77 | 116.4 | 0.597         |
| 8   | 1-2-2 | 64       | 0.816  | 202.81 | 369.1 | 1.143         |
| 9   | 1-3-2 | 62       | 0.89   | 288    | 324   | 1.441         |
| 10  | 1-4-2 | 62       | 0.871  | 184    | 348   | 1.124         |
| 11  | 1-5-2 | 59       | 0.930  | 145.19 | 234.9 | 0.819         |
| 12  | 1-6-2 | 59.5     | 0.939  | 145.19 | 234.9 | 0.819         |
| 13  | 2-1-1 | 44.5     | 0.88   | 197.4  | 499.9 | 1.22          |
| 14  | 2-2-1 | 43       | 0.91   | 186.5  | 479.0 | 1.06          |
| 15  | 2-3-1 | 53       | 0.84   | 322.28 | 687   | 1.286         |
| 16  | 2-4-1 | 56       | 0.73   | 320.8  | 820.6 | 1.409         |
| 17  | 2-5-1 | 44       | 0.877  | 331.9  | 649.2 | 1.853         |
| 18  | 2-6-1 | 56.5     | 0.718  | 320.8  | 852.1 | 1.383         |
| 19  | 2-1-2 | 61       | 0.878  | 204    | 220.5 | 1.248         |
| 20  | 2-2-2 | 64       | 0.816  | 202.81 | 369.1 | 1.143         |
| 21  | 2-3-2 | 62       | 0.871  | 184    | 348   | 1.124         |
| 22  | 2-4-2 | 62       | 0.931  | 151.88 | 184.8 | 0.839         |
| 23  | 2-5-2 | 62.5     | 0.894  | 248.57 | 269.91| 1.305         |
| 24  | 2-6-2 | 62       | 0.931  | 151.88 | 184.8 | 0.839         |

The scores of individual desirability function (d<sub>i</sub>) are calculated for each solution (h) and labelled d<sub>1</sub>, d<sub>2</sub>, d<sub>3</sub>, d<sub>4</sub>, and d<sub>5</sub> as shown in Table 10. The overall desirability function (D<sub>h</sub>) is calculated for each solution (h). From these alternative solutions scores, the fourth solution (1-4-1) shows the highest score (0.726). In this score, the assignment variables and its performance measures are considered the best alternative solution. Figure 5 shows the best set of assigned variables are straight layout, min total number of successor tasks rule, and four workers, while the results obtained concerning performance evaluation represent the lower bound of cycle time and upper bound of efficiency.
Table 10. Individual desirability function for all alternative solutions.

| NO. | Level  | d1   | d2   | d3   | d4   | d5   | Dh   |
|-----|--------|------|------|------|------|------|------|
| 1   | 1-1-1  | 1    | 0.727| 0.541| 0.434| 0.504| 0.612|
| 2   | 1-2-1  | 1    | 0.886| 0.631| 0.518| 0.631| 0.712|
| 3   | 1-3-1  | 1    | 0.978| 0.583| 0.651| 0.532| 0.723|
|     | 1-4-1  |      |      | 0.883| 0.561| 0.532| 0.726|
| 5   | 1-5-1  | 1    | 0.772| 0.234| 0.461| 0.224| 0.451|
| 6   | 1-6-1  | 1    | 0.817| 0.529| 0.483| 0.576| 0.655|
| 7   | 1-1-2  | 0.190| 0.909| 1    | 1    | 1    | 0.704|
| 8   | 1-2-2  | 0    | 0.434| 0.560| 0.656| 0.565| 0    |
| 9   | 1-3-2  | 0.095| 0.758| 0.190| 0.717| 0.328| 0.317|
| 10  | 1-4-2  | 0.095| 0.675| 0.642| 0.685| 0.580| 0.439|
| 11  | 1-5-2  | 0.238| 0.977| 0.811| 0.838| 0.822| 0.665|
| 12  | 1-6-2  | 0.214| 0.977| 0.811| 0.838| 0.822| 0.651|
| 13  | 2-1-1  | 0.928| 0.714| 0.584| 0.478| 0.504| 0.622|
| 14  | 2-2-1  | 1    | 0.846| 0.631| 0.518| 0.631| 0.705|
| 15  | 2-3-1  | 0.523| 0.537| 0.041| 0.224| 0.451| 0.260|
| 16  | 2-4-1  | 0.380| 0.052| 0.048| 0.042| 0.353| 0.108|
| 17  | 2-5-1  | 0.952| 0.701| 0    | 0.275| 0    | 0    |
| 18  | 2-6-1  | 0.357| 0    | 0.048| 0    | 0.374| 0    |
| 19  | 2-1-2  | 0.142| 0.705| 0.555| 0.858| 0.482| 0.470|
| 20  | 2-2-2  | 0    | 0.434| 0.560| 0.656| 0.565| 0    |
| 21  | 2-3-2  | 0.095| 0.675| 0.642| 0.685| 0.580| 0.439|
| 22  | 2-4-2  | 0.095| 0.943| 0.782| 0.907| 0.807| 0.552|
| 23  | 2-5-2  | 0.071| 0.779| 0.362| 0.791| 0.436| 0.370|
| 24  | 2-6-2  | 0.095| 0.943| 0.782| 0.907| 0.807| 0.552|

Figure 5. Assignment variables and performance measures for the best alternative solution calculated by T–WAWH model.

6. Conclusions and Future Works
The assembly line has been an important element of manufacturing systems in modern industries, but it needs to be reconfigured using the available resources, and performance measures enhanced to meet
the demands of rapidly changing markets. In order to demonstrate the importance of the assembly line balancing problem, a T–WAWH model is presented. This model allows decision-makers to use more important assignment variables to field ALB, namely, layout, sequence vectors and worker number. In summary, the developed model is proven to be efficient and useful to address the straight and U-shaped balancing problem, minimizing or maximizing the related criteria simultaneously. These capabilities make the proposed model significant, and useful for considering all solutions available for assembly line balancing and selecting the best solution.

However, the proposed model is limited by its restriction to straight and U-shaped assembly line balancing models, so future works could focus on solving more complicated problems such as parallel and two-sided assembly line layouts, using the T–WAWH. In addition, the proposed methodology could be solved using meta-heuristic algorithms, e.g. genetic algorithm.

References

[1] Zeqlang Z and Wenming C 2015 Improved heuristic procedure for mixed-model U-line balancing problem with fuzzy times Logistics Engineering Institution CMES (ed) China pp. 395–405.

[2] Alavidoost M, Babazadeh H and Sayyari S 2016 An interactive fuzzy programming approach for bi–objective straight and U–shaped assembly line balancing problem Applied Soft Computing, 40 pp 221–35.

[3] Uğur Ö and Bilal T 2009 A new hybrid improvement heuristic approach to simple straight and U – type assembly line balancing problems J Intell Manuf 20 pp 123–36.

[4] Venkatesh, Jonnalagedda, Balaji and Dabede 2014 Application of simple genetic algorithm to U-shaped assembly line balancing problem of type II The International Federation of Automatic Control, South Africa, pp 6118–73.

[5] Parames C and Panuwat O 2010 Mixed-model U-shaped assembly line balancing problems with coincidence memetic algorithm J. Software Engineering & Application 3 pp 347–63.

[6] Shwetank A, Rajeev J, Mishra P and Yadav H 2013 A heuristic approach for U-shaped assembly line balancing to improve labour productivity Computer & Industrial Engineering 64 pp 895–901.

[7] Mamun A, Khaled A, Ali S and Chowdhury M 2012 A heuristic approach for balancing mixed-model assembly line of type I using genetic algorithm International Journal of Production Research 50 pp 5106–16.

[8] Hadi P T, Mahsa S M, Fatemeh S and Yosef G K 2017 A multi-objective method for solving assembly line balancing problem Decision Science Letters 6 pp 1–10.

[9] Bukchin J 1998 A comparative study of performance measures for throughput of a mixed model assembly line in a JIT environment Int. J. Prod. Res. 36 pp 2669–85.

[10] Wenqiang Z and Mitsuo G 2011 An efficient multiobjective genetic algorithm for mixed-model assembly line balancing problem considering demand ratio-based cycle time J Intell Manuf 22 pp 367–78.

[11] Pasandideh S H R and Akhavan Niaki S T 2006 Optimizing multi-response statistical problem using genetic algorithm Scientia Iranica 13 pp 50–59.

[12] Liu Yang Z, Yizhong M, Linhan O and Feng W 2013 Application of the modified genetic algorithm to multi-response robust design based on entropy weight and the desirability function IEEE pp164–68.

[13] Jeong I and Kim K 2005 D-STEM: a modified step method with desirability function concept Computers & Operations Research 32 pp 3175–90.