International Journal of Membrane Science and Technology, 2024, Vol. 11, No. 1, pp 594-606

Assembly Line Balancing using the Valentine and Aseem Heuristic Algorithm (VAHA)

Chigozie Valentine Ugo-Njoku[1], Uche Joachim Opara[2]

[1]SIUE School of Engineering, Department of Industrial Engineering, Southern Illinois University, Edwadsville. Email: cugonjo@siue.edu, ORCID: https://orcid.org/0009-0003-8141-6322

[2]SIUE School of Engineering, Department of Industrial Engineering, Southern Illinois University, Edwadsville. Email: uopara@siue.edu

Abstract: This study delves into the development and application of the Valentine and Aseem Heuristic Algorithm (VAHA) as a different heuristic model to ALB problems. This was achieved by exploiting agglomerative hierarchical clustering principle, and it was proven that assembly lines have lesser idle time than the conventional line balancing methods. The metrics used for comparison were the number of stations, the balance efficiency, the total time the line was idle, and the balance delay of the line. To audit the performance and validate the VAHA model, three conventional line balancing algorithms were used to benchmark the proposed algorithm results and furthermore, the result of the VAHA model, was juxtaposed to the result of another group in the class. Upon evaluation, it shows that by applyingVAHA at the barest minimum, a reduction in total idle time at the workstations can be achieved. Moreover, this reduction in total idle time implies that overall labor efficiency has increased.

Keywords: Assembly Line Balancing, Valentine And Aseem Heuristic Algorithm, Line Efficiency, Parallel Workstation, And Total Idle Time.

LIST OF ABBREVIATIONS

ALB	Assembly Line Balancing
ALBP	Assembly Line Balancing Problem
CPM	Critical Path Method
DE	Differential Evolution
E	Line Efficiency
E_b	Balance Efficiency
GALBP	Generalized Assembly Line Balancing Problem
LE	Line Efficiency
R_p	Production Rate
RPW	Ranked Positional Weight
SALBP	Simple Assembly Line Balancing Problem
TOPSIS	Technique for Order to Preference by Similarity to Ideal Solution
T_C	Cycle Time
T_{ek}	Work Element Time
T_i	Idle Time
T_r	Repositioning Time
T_s	Service Time
T_{WC}	Total Work Content Time
UALBP	U-shaped Assembly Line Balancing Problem
VAHA	Valentine and Aseem Heuristic Algorithm
W*	Theoretical Minimum Number of Workers

1. INTRODUCTION

ALB is a critical aspect of production planning and scheduling, in manufacturing industries aimed at improving productivity and reducing waste. It involves dividing the workload among production resources to ensure most efficiency, cost-effectiveness, and minimum cycle time.

This basically involves the balanced allocation of jobs to workstations without countering the precedence relationship and maximizing a particular objective function which in most cases would be the available service time at the workstations. This problem often comes up in continuous manufacturing lines and is one of the cumbersome optimization problems. Since the investment in installing an assembly line is expensive and futuristic, it is crucial to design the line for the assembly and balance the workstations in terms of workload. Line balancing is particularly useful in industries where the production of numerous identical items is needed, such as the automotive, electronics, and food processing industries. By optimizing the flow of work through the production process, line balancing can help reduce production time, minimize idle time, and increase productivity.

The first step in line balancing is to determine the series of steps necessary to make the object. This involves breaking down the production process into a series of discrete elements (steps or tasks), each of which is allocated to a distinct workstation. Once the sequence of operations has been established, the next step is to determine the completion time of every task. After the series of operations and the completion time of all tasks have been determined, the work can be allocated to the various workstations in the manufacturing line. The goal is to allocate work elements such that each workstation has a balanced workload, with no workstation being overburdened or underutilized. This can be achieved by either moving tasks between workstations or by adjusting the time required to complete each task.

One common technique used in line balancing is the use of a precedence diagram. This is a visual representation of the production process, which shows the series of activities required to assemble the item and the dependencies between them. Activities are nodal representation and their requirement of precedence is shown by arrows linking the nodes from left to right. By analyzing the precedence diagram, it is possible to identify potential bottlenecks in the production process and to determine how best to balance the workload between workstations.

Another important aspect of line balancing is the use of cycle time analysis. This involves calculating the totality of the duration needed to assemble a single unit of the item, including all setup, processing, and finishing operations. By comparing the cycle time with the available production duration, it is possible to determine maximum service time for each workstation and to adjust the workload accordingly.

2. LITERATURE REVIEW

From the initial publication on the issue by Salveson [1] in 1955, where he proposed a linear programming solution based on the Simplex method, the study of line balancing has attracted considerable attention from academics. Other attempts at linear and integer programming by Becker and Scholl [2], have resulted in formulations with 0–1 variable that allow for the explicit representation of cycle, occurrence, and precedence constraints, but also have many variables and constraints. A lot of work, nevertheless, has gone into creating ever-better algorithms for getting those ideal solutions.

Several exact solutions to the ALB problems have been developed alongside heuristics (for example, Kilbridge and Wester [3]; Baykasoglu et al [4]; Boysen and Fliedner [5]; Chica et al [6]; Chong et al [7], Gamberini, et al. [8], Genikomsakis and Tourassis [9]; Gokeen and Erel [10]; Goncalves and Almeida [11]), dynamic programming (Kilincci [12]; Lapierre et al. [13]; Fathi et al [14]), and, more recently, genetic algorithms and tabu search are some meta-heuristic algorithms that have been developed for this purpose (Hegelson and Birnie [15]; Ozcan and Toklu [16]; Sabuncuoglu et al. [17]). Many rules of dominance or decrease, as well as bounds, help to further reinforce all of them.

In addition, as the understanding of the ALB problem expands, a plethora of scholarly works have been created on the subject, including works on the configuration planning of assembly systems and efforts to incorporate an ever-expanding set of factors from actual industrial systems. Several review articles (e.g., Kilbridge and Wester [3]; Mamun et al [18]; McMullen and Frazier [19]; Baybars [20]; Narayanan and Panneerselvam [21]; Nearchou [22]; Ozcan et al [23]) have been written to organize and synthesize the findings and make them more accessible for future researchers. In addition, other systems of categorization were applied to the vast issues already in existence. At first, researchers concentrated on the configuration's primary issue: allocating tasks to individual workstations. SALBP is a subfield of operations research due to numerous simplifying, limiting assumptions that underlie this fundamental problem.

Yet, the balancing of actual production lines will call for the incorporation of many other technological or organizational factors, which will be impactful on the form of the planning problem. Follow-up works have tried to modify the **SALBP** to account for factors like parallel stations, robotic lines, mixed/multi-product lines, alternatives for processing, cost synergies, restrictions on zoning, stochastic and series-dependent assembly duration, and human resources, all of which are present in real-world ALBP. Despite these attempts, known as the GALBP, there appears to be significant chasm in scholarly research different from real-life applications, and until recently, very few industries adopted mathematical algorithms to tackle this issue.

Type I and type II are the two categories used to classify straightforward ALBP. The required assembly tasks, task duration, cycle duration and precedence constraints are all provided for type I problems and therefore reducing the idle time at the workstations is the objective. This is because a line with fewer stations requires less space and has lower labor expenses. Type I issues typically arise when creating new manufacturing lines. To fulfill anticipated orders, fewer workstations are used. Type I problem may also be used to minimize the quantity of additional workstations that would be needed during expansion (when demand increases).

The goal of type II problems is to reduce cycle duration with a specific number of stations and manning level. The output rate was maximized in this way. Type II balancing issues typically arise when an organization aims to manufacture the maximum quantity of goods with a set number of stations and not invest in additional equipment or expanding. In this case, zoning restrictions can be seen as well as order of precedence. Subassembly lines must also be considered when balancing the main line.

Type I issues are more prevalent than type II issues. When the issue size increases, the precise algorithms that are currently accessible become infeasible. While there has been some success in developing precise or ideal approaches, there have been significant strides reported in developing heuristic or inexact approaches to the single model deterministic task times problem. Perhaps there are two factors at play here: the knowledge that there are no highly efficient ideal procedures, and the requirement to address significant issues. Priority ranking or tree search logic are used in most of the approximate methods. The approach created in this research is intended to address Type I issues. ALB issues belong to the category of combinatorial optimization problems that are NP-hard. Heuristic algorithm development has attracted a lot of interest recently and for this research, a comprehensive of literature in ALB was carried out. In each of the following subsections the review of literature on heuristic methods to solve ALBP is presented.

2.1 Single-model accurate straight Assembly Line Balancing

In this instance, where fewer workstations are needed for a specific cycle duration., was studied by Panneerselvam and Sankar [25]. They considered six heuristics from Bakar et al's [26] study and offered six original heuristics. By means of a well-planned case study, they declared that heuristics (three in number) that Dar-El contributed as well as all six of their own newly provided heuristics are the optimal heuristics for tackling single-model ALP. Future researchers can start a simulated annealing algorithm from the best result of this collection of heuristics. An effective set of heuristics provided a superior result for the ALBP, as demonstrated by Ponnambalam et al [27]. To determine the optimal collection of heuristics and reduce the quantity of stations for a given cycle duration, the known efficient range of heuristics could be supplemented with more recent faux-meta-heuristics.

Ponnambalam et al.'s [27] examination of six balancing heuristics for manufacturing processes, including Moodie and Young, RPW, Kilbridge and Wester, LCR, Z-heuristic, and Hoffman, was conducted in a comparative manner. Priority matrix, initial fit and rank after an immediate update, quantity of extra workstations, efficiency of the assembly line are all heuristic metrics that are used. By implementing an idea of task similarity in both directions for the straightforward ALBP, Khlil et al [28] improved the largest candidate rule for ALB. In this instance, the goal is creating a manufacturing line with a cycle time that is equivalent and has better work relatedness.

Aufy et al. [29] created a user-friendly internet-powered advisor for hard disk assembly line balance that takes into account several variables, with the main goal being to reduce quantity of extra workstations. The model creates an itinerary using several criteria that are included in its program. Additionally, it develops models of simulation for subscriber-specified ALBP and shows the subscriber an analysis of the recommended ALB solution. However, this interactive internet-powered counselor for balancing assembly lines used archaic heuristics.

When balancing deterministic assembly lines for a single model, Nearchou [22] took into account two criteria: reducing the duration of the cycle in a manufacturing process and delay that results from improper balance at the workstations. In the case of generic differential evolution (DE) approach, his research created an outstanding demographic heuristic to resolve this issue. Key features of this multi-purpose DE heuristic are the formulation of every ALB result's cost function as a weighted sum of numerous specific tasks and the maintenance of a separate data with several ideal solutions using pareto. Objective function components' weights, w1 and w2, are estimated using a novel self-adapted method in this work. It will be very difficult to determine which formula, when compared to that provided by Ozcan. Goken and Toklu [24], is the best.

Yeh and Kao [32] created a heuristic in both directions to divide activities among the fewest possible workstations while still maximizing the efficiency of balancing. This method combines the project management tool critical path method (CPM) with the bidirectional approach. This algorithm is contrasted with an ideal approach and a unidirectional CPM heuristic. Without conducting any statistical analysis, the comparison just counts the workstations for various literature problems. This work may be altered to increase its efficacy and address a variety of ALB issues, including straight type, parallel type, resource-constrained, and U-type assembly lines.

The goal of straightforward ALBP studied by Sotskov et al. [33] is geared towards reducing the quantity of workstations for a specific product. The manufacturing line's operations are split into two categories. Subsets 1 and 2, with Set 1 consisting of tasks with accurate task durations and Set 2 consisting of tasks with stochastic job durations. According to changes in the activity times of the second set, they investigated the reliability of the ideal outcome.

2.2 Single - Model Accurate U-type ALB

Initial study is credited to Panggabean et al [30], who suggested a formulation for dynamic programming to tackle 21 very straightforward cases (with an average of 11 tasks). For complex situations, they additionally created a heuristic method using the highest RPW. Later, Zeng et al [31] created three precise methods for the UALBP: a breadth-first branch-and-bound algorithm, a reaching dynamic programming approach and a depth-first branch-and-bound algorithm, Yegul et al. [36] looked at the U-shaped double-sided SALBP, where the goal is to discover a specific design (U-shaped and two-sided) using a first-time algorithm that minimizes total quantity of workstations for a certain cycle duration. The line is set up with crossover stations on one side in a U form, and standard straight flow stations on the other side. They suggested a several-pass arbitrary heuristic approach to determine the bare least quantity of assembly stations needed. Using a small number of literary problems, they evaluated the outcome of their model against other algorithms in use and came to the conclusion that it is effective. However, there is no statistical backing for their allegation.

With regards to U-shaped assembly line balance challenge, a heuristic-based critical path method was devised by Avikal, Jain and Mishra [37] to increase worker productivity by reducing the quantity of workstations. The initial step for this approach is to determine the set of important and unimportant tasks by assuming the assembly line is a project network and applying the CPM approach. Following that, additional momentary workstations are made by allocating the jobs and giving the most important one’s priority. If a momentary workstation's downtime is extremely small, it will be transformed into a permanent workstation. Unless all of the jobs are distributed across certain workstations, this process continues. They did not contrast this approach with any of the most effective ones already in use. The approach utilized a graph and as such, it can only be applied as a region-specific heuristic to enhance the performance of algorithms such as simulated annealing and optimization of colonies of ants.

2.3 Single-model probable straight Assembly Line Balancing

Regarding the SALBP, Liu et al. [38] developed a bidirectional heuristic with the goal of reducing cycle duration for a specified number of workstations and assembly reliability, i.e., the likelihood that the total work content time will not exceed the cycle duration for the entire manufacturing process. The jobs in this model are alternately allocated to workstations ahead. Next, the tasks are switched across workstations until the cycle duration is shortened and the assembly line reliability specified from design is met. A modified version of Moodie and Young's algorithm is compared to this algorithm, and it is discovered that the proposed heuristic was more effective.

The assembly rebalancing problem with probabilistic task times was examined by Adalberto [40]. They created a brand-new heuristic by combining the well-known Moreira et al [39] technique with a choice-making process using several criteria called "Technique for order preference by similarity to ideal solution" (TOPSIS). This study takes into account the regularly dynamic product features and sales quantity. By taking into account the reduction of two standards for efficiency, namely primary labor and predicted primary costs of incompletion, as well as job reassignments, the algorithm solves the (re)balancing problem of a production line.

In summary, the study of literature on this topic presents an extensively thorough analysis of the literature falling under single-model assembly lines. Additionally, the works of literature were categorized within specific areas according to the techniques employed to tackle the ALB problems, and conclusions are reached.

3. METHODOLOGY OF PROPOSED HEURISTIC MODEL

To actualize the goals of this research, work elements were clustered by introducing agglomerative hierarchical clustering. A proximity matrix was developed by considering the work elements as binary based on their predecessors and successors. Using the Jaccard distance as a similarity measure, and the complete (maximum) linkage method was applied to build a hierarchical clustering dendrogram and clusters with the minimum distance were merged into workstations without violating the activity precedence. Then the proposed heuristic algorithm clearly shows the proximity matrix corresponding to each iteration of the algorithm.

3.1 Method of Data Analysis

For this study, therefore, a single model assembly line was used to assemble a brand-new tiny electrical appliance. The line was opened for 15 hours a day, 250 days a year. The table below defines the work elements that have been separated out into the work content. The element timings and precedence criteria were also provided. A total of 200,000 units was produced annually. The line efficiency is estimated to be 0.96. Repositioning takes each employee 0.08 minutes. The parameters to be determined are the R_P, T_C, and w^* needed to achieve production requirements annually. The VAHA model to balance the line was developed. For the VAHA model, the efficiency of the balancing and efficiency of labor on the assembly line was determined.

Table 3.1: Task description

No.	**Element description**	**(min)**	**predecessors**
1	Place frame on work holder and clamp	0.15	–
2	Assemble fan to motor	0.37	–
3	Assemble bracket i to frame	0.21	1
4	Assemble bracket 2 to frame	0.21	1
5	Assemble motor to frame	0.58	1, 2
6	Affix insulation to bracket 1	0.12	3
7	Assemble angle plate to bracket 1	0.29	3
8	Affix insulation to bracket 2.	0.12	4
9	Attach link bar to motor and bracket 2	0.30	4. 5
10	Assemble three wires to motor	0.45	5
11	Assemble nameplate to housing	0.18	–
12	Assemble light fixture to housing	0.20	11
13	Assemble blade mechanism to frame	0.65	6, 7, 8, 9
14	Wire switch, motor. and light	0.72	10, 12
15	Wire blade mechanism to switch	0.25	13
16	Attach housing over motor	0.35	14
17	Test blade mechanism, light. etc.	0.16	15,16
18	Affix instruction label to cover plate	0.12	–
19	Assemble grommet to power cord	0.10	–
20	Assemble cord and grommet to cover plate	0.23	18. 19
21	Assemble power cord leads to switch	0.40	17, 20
22	Assemble cover plate to frame	0.33	21
23	Final inspect and remove from work holder	0.25	22
24	Package	1.75	23

The data for this problem, could be presented visually in a diagram of precedence as seen below:

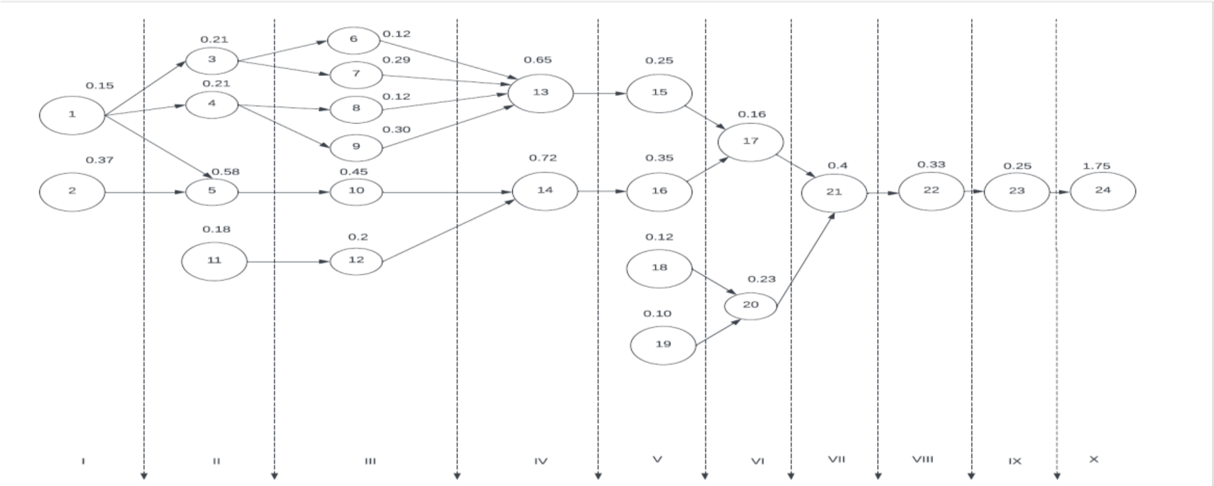


Figure 3.1: Precedence diagram

For the yearly demand, the rate of production per hour is:

$$R_p = \frac{200{,}000}{250 * 15} = 53.33\ units/hr \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots ..eqn\ 1$$

The T_c for a line efficiency of 0.96:

$$T_c = \frac{60 * 0.96}{53.33} = 1.08\ \min/cycle \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots .eqn\quad 2$$

The service time for this assembly line would be:

$$T_s = T_c - T_r\ \rightarrow 1.08 - 0.08 = \mathbf{1\ minute} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots eqn\ 3$$

Least number of workers in theory:

$$w^* = \min int \geq \frac{T_{wc}}{T_c} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots eqn\ 4$$

$$T_{WC} = \sum T_{ek} = 8.49\ mins \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots eqn\ 5$$

From eqn 4 above, $w^* = \min int\ \geq \frac{8.49}{1.08} = 7.86 \approx 8\ workers \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots eqn\ 6$

The other parameters to be used in this method and would be calculated following the application of the VAHA algorithm are:

$\sum T_S$	Total service time of all workstations
E_b	Balance efficiency
T_i	Total Idle time
d	Delay

The heuristic model that was proposed in this study is a two-stage algorithm as follows:

1) Develop an agglomerative hierarchical clustering dendrogram.

2) Allocate the clusters to parallel stations without exceeding the service time.

3.2 Developing the Valentine and Aseem Heuristic Model

The proposed VAHA algorithm would take the following flow:

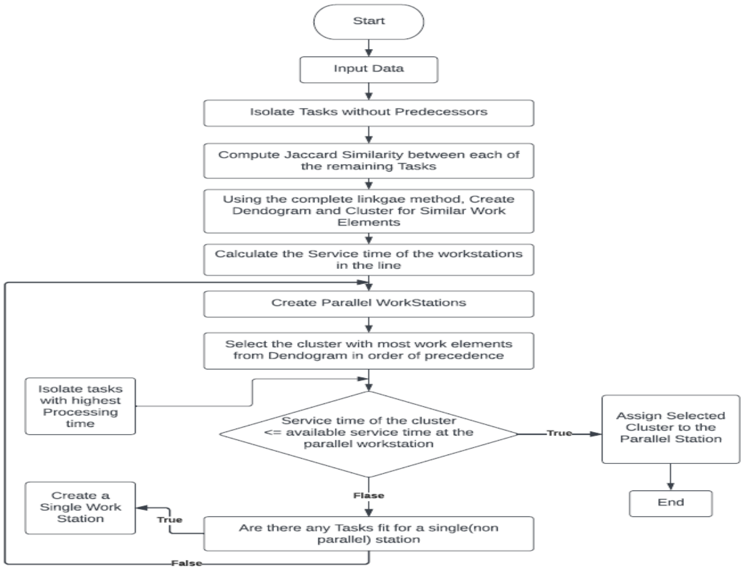


Figure 3.2: *Flowchart of proposed VAHA algorithm*

To determine the grouping, hierarchical clustering, which is a method for grouping similar data points together based on their pairwise similarity or dissimilarity was used. Specifically, the agglomerative hierarchical clustering algorithm, which considers each work element as a starting point cluster and then iteratively combines the two nearest clusters iteratively until all data points belong to the same cluster was used.

To measure the pairwise dissimilarity between two data points, the Jaccard distance, which is a ratio of the size of the corelationship to the magnitude of the combined value of the previous sets was adopted. For example, the Jaccard distance between work elements 1 and 2 is 0 because they have no common predecessors, while the Jaccard distance between work elements 4 and 5 is 0.5 because they both have work element 1 as their predecessor.

In the first step of a Jaccard Similarity measurement for two work elements, the following four quantities (i.e., frequencies) are computed for the given data:

- a = the number of predecessors that equal 1 for both elements i and j
- b = the number of predecessors that equal 0 for element i but equal 1 for element j
- c = the number of predecessors that equal 1 for element i but equal 0 for element j
- d = the number of predecessors that equal 0 for both elements i and j.

Then, Jaccard Similarity for these attributes is calculated by the following equation:

$$J(i,j) = sim(i,j) = \frac{a}{a+b+c} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \, eqn\ 7$$

The number of 0 matches is not considered important in this computation and therefore is ignored because the work elements are asymmetric binary attributes. The data was then represented as a dissimilarity matrix, where the (i,j)-th entry is the Jaccard distance between data points i and j. These values were calculated using Jaccard distance metric with a similarity threshold of 0.5.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0.8	1	1	0.4	1	1	1	0.6	1	0.7	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0.8	0	1	1	0.3	1	1	1	0.7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	0	0	0.8	0.5	0.7	0.7	1	1	1	1	0.3	1	1	1	1	1	1	1	1	1	1	1
4	1	1	0	0	0.8	0.5	0.7	0.7	1	1	1	1	0.3	1	1	1	1	1	1	1	1	1	1	1
5	0.4	0.3	0.8	0.8	0	1	1	1	0.8	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	0.5	0.5	1	0	0.3	0.3	1	1	1	1	0.8	0.8	1	1	1	1	0.5	0.8	1	1	1	1
7	1	1	0.7	0.7	1	0.2	0	0.7	0.7	1	1	1	0.8	1	1	1	1	1	1	1	0.5	1	1	1
8	1	1	0	0.5	1	0.5	0.5	0	0.4	0.4	1	1	1	1	0.1	0.5	0.5	1	1	1	1	1	1	1
9	0.8	1	1	0.4	1	1	1	0.6	0	1	0.7	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	0.3	1	1	1	0.7	1	1	0	0.3	1	1	1	1	0.8	0.8	1	1	1	1	0.5	0.8	1
11	0.7	1	1	1	1	0.5	1	1	1	1	0	1	0.5	1	0	1	0.8		1	0.5	0.3	1	0.7	0.7
12	0.8	1	1	1	1	0.9	1	1	1	0.8	0	0	0.6	0.7	1	0.7	0.9	1	1	0.9	1	1	1	1
13	0.8	1	1	1	1	0.7	0.7	0.7	0.5	0.3	1	0.8	0	0.3	0.8	0.5	0.6	0.6	1	0.5	0.4	0.3	0.5	0.1
14	1	1	0.7	0.8	0.7	0.8	0.8	0.8	0.8	0.5	1	0.8	0.6	0	0.9	0.8	1	0.8	1	1	0.8	1	0.9	1
15	1	1	1	1	1	1	1	0.1	1	1	0	1	0.8	0.9	0	1	0.5	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	0.5	1	0.8	1	0.7	0.5	0.8	0	0.5	0.9	0.7	0.9	0.3	0.5	0.5	0.8
17	1	1	1	1	1	1	1	0.5	1	0.8	0.8	0.9	0.6	1	0.5	0.5	0	1	0.5	0.7	1	0.8	0.4	0.7
18	1	1	1	1	1	1	1	1	1	1		1	0.6	0.8	1	0.9	1	0	1	0.3	0.5	0.5	1	0.4
19	1	1	1	1	1	0.5	1	1	1	1	1	1	1	1	1	0.7	0.5	1	0	0.5	0.8	0.3	0.2	0.6
20	1	1	1	1	1	0.8	1	1	1	1	0.5	0.9	0.5	1	1	0.9	0.7	0.3	0.5	0	0.7	1	0.7	0.9
21	1	1	1	1	1	1	0.5	1	1	1	0.3	1	0.4	0.8	1	0.3	1	0.5	0.8	0.7	0	1	0.8	0.3
22	1	1	1	1	1	1	1	1	1	0.5	1	1	0.3	1	1	0.5	0.8	0.5	0.3	1	1	0	0.2	0.4
23	1	1	1	1	1	1	1	1	1	0.8	0.7	1	0.5	0.9	1	0.5	0.4	1	0.2	0.8	0.8	0.2	0	0.5
24	1	1	1	1	1	1	1	1	1	1	0.7	1	0.1	1	1	0.8	0.7	0.4	0.6	0.3	0.3	0.4	0.5	0

Table 3.2: *Similarity distance between the work elements*

Using this matrix, the agglomerative hierarchical clustering algorithm using the complete linkage method to group the data points into a dendrogram, which is a tree-like diagram that shows the hierarchy of the clusters as seen in the figure below can be applied.

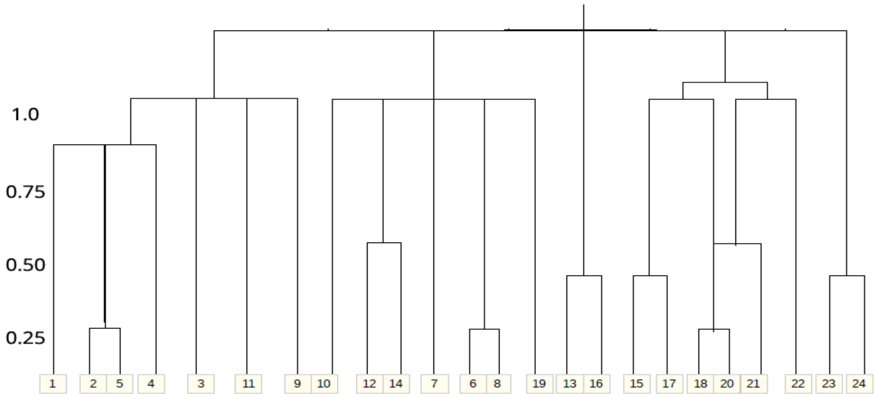
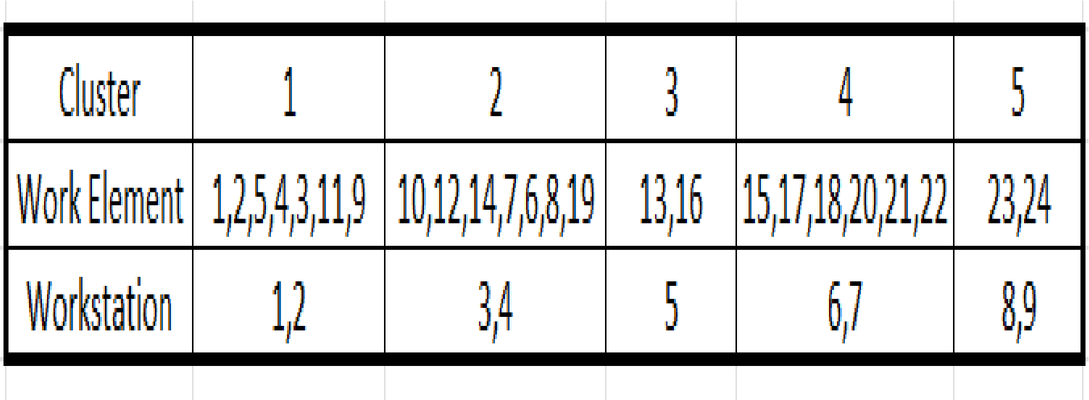


Figure 3.3: Dendrogram of agglomerative clustering by complete (max) linkage

The complete (max) linkage method of agglomerative clustering is given by:

$$D_{ij} = \max\{d(x_i, x_j), x_i \in C_i, x_j \in C_j\} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots.. eqn\ 8$$

This method of linkage was chosen, because it is less susceptible to noise and outliers and as such it would have:



Cluster	1	2	3	4	5
Work Element	1,2,5,4,3,11,9	10,12,14,7,6,8,19	13,16	15,17,18,20,21,22	23,24
Workstation	1,2	3,4	5	6,7	8,9

Table 3.3: Clustering by complete (max) linkage method

To obtain the desired parallel workstations, one can cut the dendrogram at appropriate levels that separate the clusters into five disjoint subsets by considering the maximum cophenetic distance, which is the largest pairwise dissimilarity in any two data points in a specific cluster. Specifically, levels that maximize the minimum cophenetic distance between any two clusters that belong to different groups were selected.

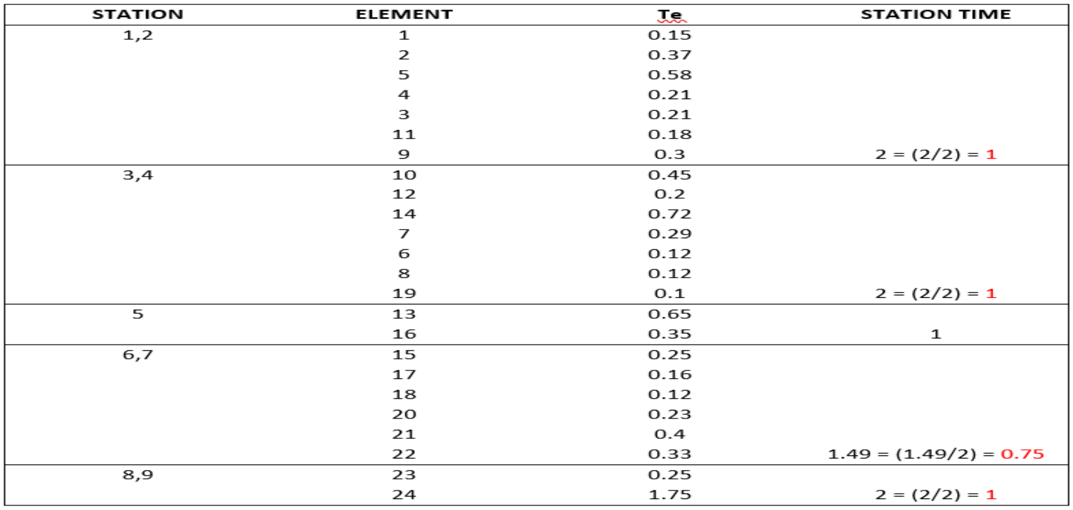
Using this approach of allocating the activities, the study began with activities in the initial cluster of the dendrogram, allocate the activities to parallel stations without exceeding a calculated service time of one minute. Interesting to note that the fifth workstation is not parallel. It became immediately evident, in addition to the enhanced efficiency, another advantage of this approach is that it could produce a sequential allocation of activities to the workstations as soon as the dendrogram is generated, because of having high performance power in solving the precedence constraints.

4. COMPUTATIONAL RESULT

4.1 Indicators of Performance

Results of proposed VAHA model are compared with several conventional algorithms for the same problem and also compared with the solutions that were obtained by another group in the class – the maximum path method by Kongara Lakshman Kumar and Suneesh Gummadi.

The allocation of work elements to their respective workstations using the VAHA method resulted in:



STATION	ELEMENT	Te	STATION TIME
1,2	1	0.15	
	2	0.37	
	5	0.58	
	4	0.21	
	3	0.21	
	11	0.18	
	9	0.3	2 = (2/2) = 1
3,4	10	0.45	
	12	0.2	
	14	0.72	
	7	0.29	
	6	0.12	
	8	0.12	
	19	0.1	2 = (2/2) = 1
5	13	0.65	
	16	0.35	1
6,7	15	0.25	
	17	0.16	
	18	0.12	
	20	0.23	
	21	0.4	
	22	0.33	1.49 = (1.49/2) = 0.75
8,9	23	0.25	
	24	1.75	2 = (2/2) = 1

Table 4.1: Allocation of work elements to workstation using VAHA model

To validate the VAHA model's accuracy, the result presented above will be analyzed using these metrics:

Balance Efficiency (E_b):

Total idle time = 0.25 mins

Total workstation time = 9 mins

$$d = \frac{Idle\ time}{Total\ workstation\ time} = \frac{0.25}{9} = 0.028 \approx 2.8\% \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots eqn\ 9$$

$$E_b = 1 - 0.028 = 0.972 \approx \mathbf{97.2}\% \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots eqn\ 10$$

Overall Labor Efficiency on the Line (E_L) = $E * E_r * E_b \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots eqn\ 11$

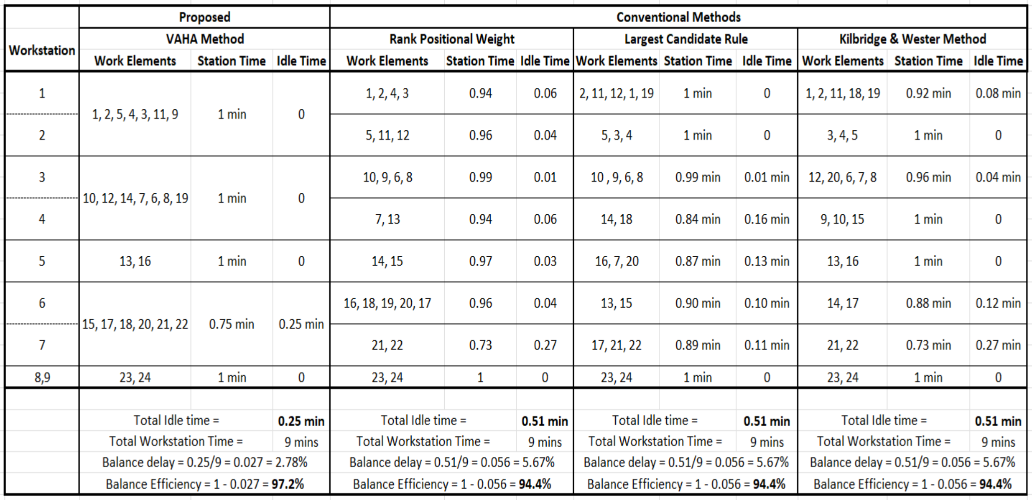
$$Line\ efficiency\ (E) = 0.96 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots eqn\ 12$$

$$Repositioning\ Efficiency\ (E_r) = \frac{T_S}{T_C} = \frac{1}{1.08} = 0.93 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots eqn\ 13$$

$$From\ eqn\ 11, E_L = 0.96 * 0.93 * 0.97 = 0.866 \approx \mathbf{86.6}\%$$

4.2 Benchmarking (Performance Comparison with well-known heuristic models):

The proposed method would then be used to compare with the three conventional assembly line balancing algorithms that were taught in class; namely Rank Positional Weight, LCR, Kilbridge and Wester algorithm. It is important to note that this comparison is based on a specific metric i.e., balance delay, using each of these heuristic algorithms as shown in the table below:

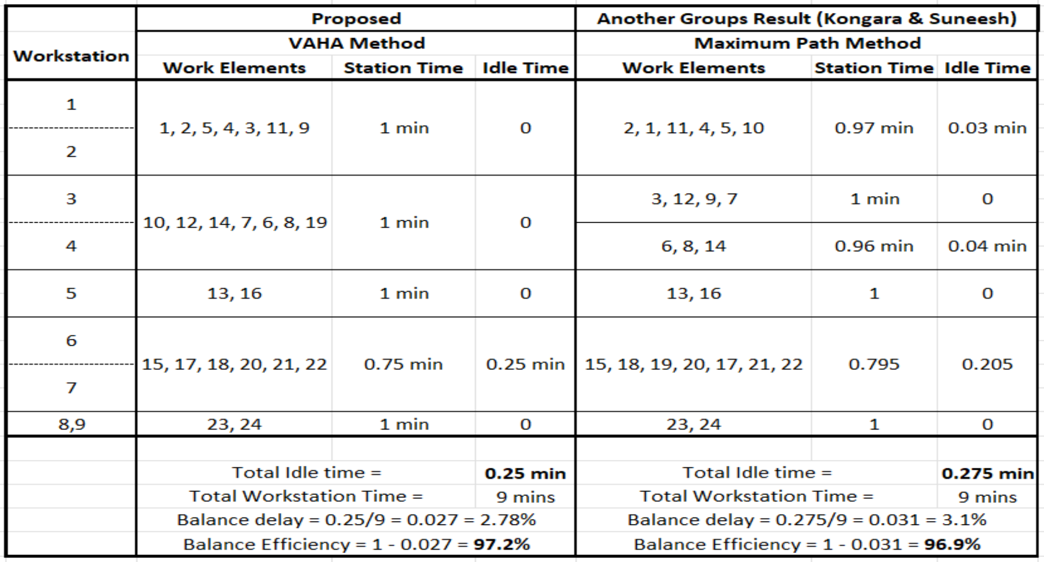


Workstation	Proposed			Conventional Methods								
	VAHA Method			Rank Positional Weight			Largest Candidate Rule			Kilbridge & Wester Method		
	Work Elements	Station Time	Idle Time	Work Elements	Station Time	Idle Time	Work Elements	Station Time	Idle Time	Work Elements	Station Time	Idle Time
1	1, 2, 5, 4, 3, 11, 9	1 min	0	1, 2, 4, 3	0.94	0.06	2, 11, 12, 1, 19	1 min	0	1, 2, 11, 18, 19	0.92 min	0.08 min
2				5, 11, 12	0.96	0.04	5, 3, 4	1 min	0	3, 4, 5	1 min	0
3	10, 12, 14, 7, 6, 8, 19	1 min	0	10, 9, 6, 8	0.99	0.01	10 , 9, 6, 8	0.99 min	0.01 min	12, 20, 6, 7, 8	0.96 min	0.04 min
4				7, 13	0.94	0.06	14, 18	0.84 min	0.16 min	9, 10, 15	1 min	0
5	13, 16	1 min	0	14, 15	0.97	0.03	16, 7, 20	0.87 min	0.13 min	13, 16	1 min	0
6	15, 17, 18, 20, 21, 22	0.75 min	0.25 min	16, 18, 19, 20, 17	0.96	0.04	13, 15	0.90 min	0.10 min	14, 17	0.88 min	0.12 min
7				21, 22	0.73	0.27	17, 21, 22	0.89 min	0.11 min	21, 22	0.73 min	0.27 min
8,9	23, 24	1 min	0	23, 24	1	0	23, 24	1 min	0	23, 24	1 min	0
	Total Idle time =		**0.25 min**	Total Idle time =		**0.51 min**	Total Idle time =		**0.51 min**	Total Idle time =		**0.51 min**
	Total Workstation Time =		9 mins	Total Workstation Time =		9 mins	Total Workstation Time =		9 mins	Total Workstation Time =		9 mins
	Balance delay = 0.25/9 = 0.027 = 2.78%			Balance delay = 0.51/9 = 0.056 = 5.67%			Balance delay = 0.51/9 = 0.056 = 5.67%			Balance delay = 0.51/9 = 0.056 = 5.67%		
	Balance Efficiency = 1 - 0.027 = **97.2%**			Balance Efficiency = 1 - 0.056 = **94.4%**			Balance Efficiency = 1 - 0.056 = **94.4%**			Balance Efficiency = 1 - 0.056 = **94.4%**		

Table 4.2: Comparative analysis of efficiency of the balancing

4.3 Comparing with the Result Generated by Another Group:

The VAHA model was also contrasted to the outcome generated by another group. In prioritizing the work elements with the highest processing time along its path through a method they called maximum path method, Kongara Lakshman Kumar and Suneesh Gummadi generated a result for balancing the same assembly line problem. Comparing their result with the result of the proposed VAHA model, is seen below:



Workstation	Proposed			Another Groups Result (Kongara & Suneesh)		
	VAHA Method			Maximum Path Method		
	Work Elements	Station Time	Idle Time	Work Elements	Station Time	Idle Time
1	1, 2, 5, 4, 3, 11, 9	1 min	0	2, 1, 11, 4, 5, 10	0.97 min	0.03 min
2						
3	10, 12, 14, 7, 6, 8, 19	1 min	0	3, 12, 9, 7	1 min	0
4				6, 8, 14	0.96 min	0.04 min
5	13, 16	1 min	0	13, 16	1	0
6	15, 17, 18, 20, 21, 22	0.75 min	0.25 min	15, 18, 19, 20, 17, 21, 22	0.795	0.205
7						
8,9	23, 24	1 min	0	23, 24	1	0
	Total Idle time =		**0.25 min**	Total Idle time =		**0.275 min**
	Total Workstation Time =		9 mins	Total Workstation Time =		9 mins
	Balance delay = 0.25/9 = 0.027 = 2.78%			Balance delay = 0.275/9 = 0.031 = 3.1%		
	Balance Efficiency = 1 - 0.027 = **97.2%**			Balance Efficiency = 1 - 0.031 = **96.9%**		

Table 4.3: Comparison of results of VAHA model and Maximum path method

The results above clearly show that the VAHA model gives a better balance efficiency than the maximum path method.

CONCLUSION

For this project, the ALBP was solved with a goal of maximizing the utilization of every workstation – having the station time as close as possible to the available service time and reducing total idle time in the assembly line. Primarily, a heuristic model that considers the work elements as binary attributes based on their predecessors and successors was proposed and the distances between all the work elements was established and using the complete or maximum linkage method, the work elements were clustered in a dendrogram. Following this proposed heuristic model, parallel workstations were implemented in accordance with every cluster in the dendrogram.

In establishing the reliability of our VAHA model, our results were juxtaposed with that of conventional heuristic algorithms. The comparisons show that our VAHA model performed better than the conventional heuristic algorithms in terms of balance efficiency and total idle time reduction.

Furthermore, we also juxtaposed the outcome of the VAHA model with the algorithm established by a different group in the class. The comparison shows that our proposed model has better balance efficiency and less total idle time compared to the model proposed by the other group.

The success of our proposed heuristic model is attributed to the incorporation of a measure of similarity distance, which takes care of the precedence constraint by default and clusters similar work elements into a parallel workstation with respect to the service time. This modification makes the proposed heuristic model particularly helpful in real-life manufacturing industries by increasing efficiency because utilizing multiple workstations simultaneously, would provide greater flexibility and changes in the manufacturing process can be accommodated more easily to improve responsiveness to customer demands as well as adapt to ever-dynamic market conditions, significantly reduce the time taken to complete each task, leading to faster production times and increased productivity. Moreover, because the proposed heuristic model focuses on reducing the overall idle time, the manufacturing/assembly process becomes more streamlined, resulting in faster production times and increased output. An improvement to the quality control of the manufacturing process can also be achieved using the proposed model because by reducing idle time, quality control measures can be more effectively implemented, resulting in higher-quality products.

The strength of the VAHA model is that it takes care of the precedence constraints by default as soon as the measure of similarity distance is determined and the dendrogram is developed. In contrast, this would imply that the disadvantage of the proposed model is that it is complex and takes longer computational time.

Lastly, future scholarly studies based on our solution in this project would possibly be to analyze the effect of using other measures of similarity such as the Manhattan and Euclidean distances to develop a similarity matrix and then using different linkage methods such as either single (minimum) linkage, average linkage, or centroid linkage methods to develop the dendrogram. It would be interesting to see if the model would still accurately cluster the work elements into parallel stations and not going against the constraints of precedence.

References

[1] Salveson, M. E. (1955). "The Assembly Line Balancing Problem," American *Journal of Industrial Engineering*, May-June 1955 Issue, pp. 7-12.

[2] Becker C. & Scholl A. (2006). "A survey on problems and methods in generalized assembly line balancing," *European Journal of Operational Research,* vol. 168(3), pp. 694-715.

[3] Kilbridge, Maurice D, and Leon Wester (1962) "A review of Analytical Systems of Line Balancing." *Operations research* 10.5, pp 626-638.

[4] Baykasoglu A, Dereli T (2008). “Two-sided assembly line balancing using an ant-colony-based heuristic.” *International Journal of Advanced Manufacturing Technology* Vol 36: pp 582–588.

[5] Boysen N, Fliedner M (2008). “A versatile algorithm for assembly line balancing.” *European Journal of Operations Research*, Vol. 184, pp 39–56.

[6] Chica M, Cordon O, Damas S, Bautista J (2010). “Multi-objective constructive heuristics for the1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search.” *Journal of Information Sciences*, Vol 180, pp 3465–3487.

[7] Chong K.E, Omar M.K, Baker N.A (2008). “Solving assembly line balancing problem using genetic algorithm with heuristic treated initial population.” *Proceedings of the World Congress on Engineering* ISBN:978-988-17012-3-7

[8] Gamberini R, Grassi A, Rimini B (2006). “A new multi-objective heuristic algorithm for solving the stochastic line re-balancing problem.” *International Journal of Production Economics*, Vol 102, pp 226–243.

[9] Genikomsakis K.N, Tourassis V.D (2010). “Enhancing the largest candidate rule for assembly line balancing through the concept of bi-directional work relatedness.” *Journal of Advanced Computer Intelligence Information*. 14(4):353–363

[10] Gokcen H, Erel E (1998). “Binary integer formulation for mixed model assembly line balancing problem”. *Journal of Industrial Engineering*, Vol 34(2), pp 451–461.

[11] Goncalves J.F, Almeida J.R.D (2002). “A hybrid genetic algorithm for assembly line balancing”. *Journal of Heuristics*, Vol 8, pp 629–642.

[12] Kilincci O (2010). “A Petri net-based heuristic for simple assembly line balancing problem of type 2.” *International Journal of Advanced Manufacturing Technology*, Vol. 46, pp 329– 338.

[13] Lapierre S.D, Ruiz A, Soriano P (2006). “Balancing assembly lines with tabu search.” *European Journal of Operations Research,* Vol.168, pp 826–837.

[14] Fathi, M., Alvarez, M. J. & Rodríguez, V. (2011). "A New Heuristic Approach to Solving U-shape Assembly Line Balancing Problems Type-1," *World Academy of Science,* vol. 5, pp. 269-277.

[15] Helgeson W.P. & Birnie D. P. (1961). "Assembly Line Balancing Using the Ranked Positional Weight Technique," *Journal of Industrial Engineering*, vol. 6, pp. 394-398.

[16] Ozcan, U. & Toklu, B. (2009). "A new hybrid improvement heuristic approach to simple straight and U-type assembly line balancing problems," Intelligent Manufacturing, vol. 20, pp. 123-136.

[17] Sabuncuoglu I, Erel R, Tanyer M (2000). “Assembly line balancing using genetic algorithm.” *Journal of Intelligent Manufacturing,* Vol. 11, pp 295–310.

[18] Mamun A.A, Khaled A.A, Ali S.M, Chowdhury M.M (2012). “A heuristic approach for balancing mixed-model assembly line of type I using genetic algorithm.” *International Journal of Production Research*, Vol 50, pp 5106–5116.

[19] McMullen P.R, Frazier G.V (1998). “Using simulated annealing to solve a multi-objective assembly line balancing problem with parallel workstations.” *International Journal of Production Research*, Vol 36(10), pp 2717–2741.

[20] Baybars, I., (1986). “An efficient heuristic method for the simple assembly line balancing problem,” *International Journal of Production Research,* Vol. 24, No. 1, pp. 149-166.

[21] Narayanan S, Panneerselvam R (2000). “New efficient set of heuristics for assembly line balancing.*” International Journal of Management Systems,* Vol. 16(3), pp 287–300.

[22] Nearchou A.C (2008). “Multi-objective balancing of assembly lines by population heuristics.” *International Journal of Production Research*, Vol. 46(8), pp 2275–2297.

[23] Ozcan U, Cercioglu H, Gokcen H, Toklu B (2009). “A tabu search algorithm for the parallel assembly line balancing problem.” *Gujarat Journal of Science,* Vol. 22(4), pp 313–323.

[24] Ozcan U, Gokcen H, Toklu B (2010). “Balancing parallel two-sided assembly lines.” *International Journal of Production Research*, Vol. 48(16), pp 4767–4784.

[25] Panneerselvam R, Sankar CO (1993). “New heuristics for assembly line balancing problem.” *International Journal of Manufacturing Systems,* Vol 9(1), pp 25–36.

[26] Bakar, Ramli, M. F., Zakaria, M. Z., Sin, T. C., & Masran, H. (2018). “Minimization of Bottleneck and Workstations for Assembly Line Balancing Problem in Power Transformer Manufacturing Using Heuristics”. *IEEE 5th*

International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), pp 1–5. https://doi.org/10.1109/ICSIMA.2018.8688780

[27] Ponnambalam SG, Aravindan P, Naidu GM (1999). “A comparative evaluation of assembly line balancing heuristics.” *Institute of Advanced Manufacturing Technology,* Vol. 15, pp 577–586.

[28] Khlil, Al-Khazraji, H., & Alabacy, Z. (2020). “Solving Assembly Production Line Balancing Problem Using Greedy Heuristic Method.” *IOP Conference Series. Materials Science and Engineering*, *745*(1), 12068–. https://doi.org/10.1088/1757-899X/745/1/012068

[29] Aufy, Samah A, and AllaEldin H Kassam (2020). “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.” *IOP conference series. Materials Science and Engineering* 671.1: 12147–. Web.

[30] Panggabean, Barjaman, M. R, & Nugraha, S. P. (2018). “Assembly Line Balancing Using Heuristic Methods.” *Advances in Natural and Applied Sciences, Vol.* 12(8), 1–. https://doi.org/10.22587/anas.2018.12.8.1.

[31] Zeng, Liu, L., Shen, L., & Wu, L. (2014). “Heuristic balancing schedule method for complicated assembly line of single product.” *Ji suan ji gong cheng yu ying yong, 50(7),* pp 263–270. https://doi.org/10.3778/j.issn.1002-8331.1204-0716.

[32] Yeh D.H, Kao H.H (2009). “A new bidirectional heuristic for the assembly line balancing problem.” *Computerized Industrial Engineering Journal,* Vol 57, pp 1155–1160.

[33] Sotskov Y.N, Dolgui A, Portmann M.C. (2006). “Stability analysis of an optimal balance for an assembly line with fixed cycle time.” *European Journal of Operations Research*, Vol. 168, pp 783–797.

[34] Scholl, Boysen, N., & Fliedner, M. (2013). “The assembly line balancing and scheduling problem with sequence-dependent setup times: problem extension, model formulation and efficient heuristics.” *Operations Research Spectrum,* Vol. 35(1), pp 291–320. https://doi.org/10.1007/s00291-011-0265-0

[35] Huskhazrin Kharuddin, Fadzli Ramli, M., & Hafiz Masran, M. (2020). “Line balancing using heuristic procedure and simulation of assembly line.” *Indonesian Journal of Electrical Engineering and Computer Science*, Vol.17(2), pp 774. https://doi.org/10.11591/ijeecs.v17.i2.pp774-782

[36] Yegul M.F., Agpak K, Yavuz M (2010) “A new algorithm for U-shaped two-sided assembly line balancing.” *Trans Can Society of Mechanical Engineering,* Vol. 34(2), pp 225–241.

[37] Avikal S, Jain R, Mishra PK, Yadav HC (2013) “A heuristic approach for U-shaped assembly line balancing to improve labour productivity.” *Computerized Industrial Engineering,* Vol. 64, pp 895–901.

[39] Moreira, Miralles, C., & Costa, A. M. (2015). “Model and heuristics for the Assembly Line Worker Integration and Balancing Problem.” *Computers & Operations Research,* Vol. 54, pp 64–73. https://doi.org/10.1016/j.cor.2014.08.021

[40] Adalberto Sato Michels. (2023). “Model and Heuristics for the Multi-manned Assembly Line Worker Integration and Balancing Problem [Data set].” *Mendeley.* https://doi.org/10.17632/hhhcdt9c5d.1

DOI: https://doi.org/10.15379/ijmst.v11i1.3729