Solving Assembly Production Line Balancing Problem Using Greedy Heuristic Method

S Khlil¹*, H Al-Khazraji² and Z Alabacy²

¹ Technical College-Baghdad, Middle Technical University, Baghdad, Iraq, E-mail: sohaib-khlil@mtu.edu.iq.
² Control and System Engineering Department, University of Technology, Baghdad, Iraq

Abstract. This paper focuses on assembly line balancing (ALB) problem which the objective is to maximize the assembly line efficiency. The problem solved using a greedy heuristic method. MATLAB Software is used to perform the proposed greedy heuristic method. Then, the proposed method is applied to a previous real life case problem that is found in literature for the cookers assembly line in the Light Industrial Company in Iraq. The results of the proposed method is compared with the performance of the company without applying any assembly line balancing method. The result of this paper is also compared with the results obtained from a previous work in the literature which is based on Shortest Operation Time (SOT) using the software named Quantitative Methods, Production and Operations Management (POM-QM). The outcomes of the research have shown that the greedy heuristic method is more efficient, where the efficiency increased from 78.24% before applying assembly line balancing and 81.64% for the previous work based on SOT method to 85.53% when applying the greedy heuristic method. The research has recommended that the decision maker in the company should follow the most appropriate method to achieve a high efficiency operations.

1. Introduction

The assembly line balancing (ALB) problem is a decision making problem of grouping tasks required to assemble a product among set of workstations with respect to some constrains and objectives [1]. In other word, a series of workstations and the transport mechanism (convoyer belt) is referred as the assembly line. The parts that need to be assembled are processed as set of tasks, for a given cycle time. The cycle time is the time required in the stations so the stations fulfil their tasks before passing on the work piece to the next following station. Each workstation is sequenced according to the task precedence relationships. The problem here is how the tasks should be assigned to workstations such that specific objectives, including minimizing the cycle time for a given number of workstations, minimizing the number of workstations for a given cycle time, or maximizing the efficiency of the assembly line subject to the precedence relationships among tasks, are satisfied [2]. Another class of this problem is to considered different objectives at the same time. Cerqueus and Delorme [3] minimized the takt time and the number of stations. Takt time is the maximum time that the product is need be produced depend on the customer demand.

The ALB can be divided into two general categories: Simple Assembly Line Balancing (SALB) problems and Generalized Assembly Line Balancing (GALB) problems [4]. When a sequence assembly line is processed a unique model of a single-product has known and certain value of all its
input parameters, in this case the problem is named SALB problems. On other hand, GALB problems deal with all of the problems that are not considered as SALB problems (i.e. balancing of mixed model, U-shaped, parallel, and two-sided lines with stochastic-dependent processing times) [5]. Position and/or accessibility constraints are recent topic in the context of ALB. For example, Seçmen and Özbakır [6] considered a case where the assembling product cannot be moved because of the size of the assembled parts of the product.

The ALB problem could be found in numerous industries such as the home appliance, automotive, or electronics, where the main goal is to produce and deliver large amounts of standardized products efficiently [7]. Since the ALB problem is considered as NP hard class of combinatorial optimization problems [8], several methods have been developed in the literature to provide an exact solution or near optimal solution. These methods include exact methods, simple heuristic rule and meta-heuristics [1]. The historical development of Assembly Line Balancing (SALB) problems is dating back to the times of Henry Ford. In that time, assembly lines were developed for a cost efficient mass-production of standardized products [9]. Salveson [10] formulated the mathematical model of the problem as a linear programming (LP) for the first time. Over the years, the SALB has been extensively examined in the literature. In the early studies, many researchers have been adopted exact approaches for solving SALB which can be subdivided into dynamic programming (DP) and branch and bound (B&B). The first DP procedure for solving SALB was introduced by [11]. On other hand, a considerable number of B&B approaches have been developed in the literature. For example, Jackson [12] introduced a depth-first B&B algorithm named FABLE for solving SALB. Another depth-first B&B algorithm named EUREKA was proposed by [13]. Later, Scholl and Klein [14] developed a bi-directional B&B algorithm named SALOME for solving SALB.

Besides exact methods, large varieties of heuristic with different characteristics approaches to solve SALB have been proposed. Helgeson and Birnie [15] introduced the recognized Ranked Positional Weight Technique (RPWT). Kilbridge and Wester [16] developed a heuristic procedure that selects an element for assignment tasks to workstations depend on their location in the precedence diagram. Agrawal [17] introduced a heuristic procedure based on decision rule called Largest Set Rule (LSR). The tasks are selected for assigning to stations on the basis of the cumulative time for each task. The cumulative time of each task is the time required to perform the task and all the tasks preceding it. The third class of approaches is the meta-heuristic methods including the Simulated Annealing (SA) [18], the Tabu Search (TS) [19], the Genetic Algorithms (GA) [20], and the Ant Colony Optimization (ACO) [3]. In this paper, a greedy heuristic method is proposed in this paper to maximise the assembly line efficiency of the SALB. MATLAB Software is used to perform the proposed greedy heuristic method. Despite the power of the solution found by using different methods, simple heuristics are still the most commonly used tools in industry [21]. Therefore, a greedy heuristic method is proposed in this paper to maximise the assembly line efficiency. Then, the proposed method is applied to a previous real life case problem found in literature for the cookers production line in the Light Industrial Company in Iraq. The results of the proposed method are compared with the performance of the company without applying any assembly line balancing method. The result of this paper is also compared with the results obtained from a previous work in the literature which is based on Shortest Operation Time (SOT) using the software named Quantitative Methods, Production and Operations Management (POM-QM).

The remaining of this paper is organized as follows: the problem description is given in Section 2. Section 3 presents the proposed greedy heuristic method. Section 4 contains an application of the proposed heuristic method to a previous real life case problem found in literature for the cookers production line in the Light Industrial Company in Iraq. Section 5 presents the results. Section 6 discusses the results. Section 7 summarises and concludes the work.

2. Problem Description

The SALB problem is concerned with single product assembly lines where only precedence constraints between tasks are considered. In the SALB problem, a set of tasks \( (v_i) (i = 1, \ldots, n) \) need to be assigned into workstations \( (W_j) (j = 1, \ldots, m) \) subjected to assembly constraints and predefine objectives. The time required to complete a single task \( (v_i) \) is named task time \( (t_i) \). Meanwhile, the
time required to complete all the tasks within a workstation \(W_j\) is named as process time \(p_j\) and it given by [22]:
\[
P_j = \sum_{i \in W_j} t_i
\]
(1)
The highest \(p_j\) among all workstations is called as cycle time \(c\) as follows [22]:
\[
c = \text{Max}\{p_j\}, \ j = 1, \ldots, m
\]
(2)
As the cycle time is the highest process time among all workstations, therefore the idle time \(IT\) which is unproductive time resulted from the difference between cycle time and processing time of each station and can be calculated as [22]:
\[
IT = (m \times c) - \sum_{i=1}^{n} t_i
\]
(3)
Line efficiency \(\eta\) is calculated from the ratio between total processing time in all workstations divided by the multiplication of cycle time \(c\) and number of workstation \(m\) as follows [22]:
\[
\eta = \frac{\sum_{i=1}^{n} t_i}{c \times m}
\]
(4)
The sequential relationships among tasks in the SALB problem can be represented in a precedence diagram as shown in the practical case study of this paper which is given in Figure 1. This diagram contains of numbers of circles and arrows, where the circle refers to the individual task and the arrow reflects the successor tasks [5]. The SALB problem can be divided into three groups according to the objectives [1]:
- SALB-1: the objective is to minimise the number of stations, given cycle time.
- SALB-2: the objective is to minimise the cycle time, given number of stations.
- SALB-E: the objective is to maximise the efficiency of the line for variable cycle time and number of stations.

In this paper, maximise the efficiency of the line is considered as the objective that need to be improved.

**Table 1. Greedy heuristic algorithm**

| Input: |
| --- |
| • Number of task \(n\) with task time \(t_i\). |
| • Number of workstation \(m\) |

Calculate theoretical cycle time \(c_{th}\) using equation (5)

**Repeat:**

For \(i = 1\) to number of task \(n\)

• Assign task \(v_i\) to workstation \(W_j\) based on task precedence diagram as shown in Figure 1 and compute the accumulated process time \(p_j\) of each workstation.

• Compute the difference time between the theoretical cycle time \(c_{th}\) and the accumulated process time \(p_j\) of two groups: \(df_i\) the one that is less than theoretical cycle time with task \(v_{i-1}\) and \(df_2\) the one that is more than theoretical cycle time with task \(v_i\).

• If \(df_1 \leq df_2\) then add just task \(v_{i-1}\) to the workstation \(W_j\); otherwise add just task \(v_i\) to the workstation.

End

**Output:**

Print the best solution found
3. Greedy Heuristics Approach

This greedy heuristics approach aimed to increase the efficiency of the SALB by reducing the difference between the process times \( p_j \) of each workstation and the highest \( p_j \) among all workstations (cycle time). The general procedure for solving SALB can be described as follows:

1- Construct a characteristic table to illustrate the number of activities, a brief description of each activity, predecessor activity, and the processing time for each activity.

2- Construct the precedence diagram to illustrate the sequential relationships among activities.

3- Determine the theoretical cycle time \( c_{th} \) required which is calculated by dividing the total activity time by the required number of work stations \( m \) as follows:

\[
c_{th} = \frac{\sum_{i=1}^{n} t_i}{m}
\]  

4- Assign activities to a workstation using greedy heuristics approach as shown in Table (1) to find the most appropriate activity among set of activities compatible with activities that is already assigned until the station is filled.

5- Calculate process times \( p_j \) of each workstation and the highest \( p_j \) among all workstations is considered the cycle time.

6- Calculate the idle time and line efficiency.

| Activity Number | Activity Description                          | Activity Name | Time (min) | Activity that must Predecessor |
|-----------------|----------------------------------------------|---------------|------------|-------------------------------|
| 1               | Upper Roasting Part is assembly for oven      | A             | 1.5        |                               |
| 2               | Lower Roasting Part is assembly for oven      | B             | 2          |                               |
| 3               | Oven Frame is placed on conveyor             | C             | 0.5        | A, B                         |
| 4               | Upper Roasting Part is attached              | D             | 1.5        | C                             |
| 5               | Lower Roasting Part is attached              | E             | 2.5        | D                             |
| 6               | Thermal Insulator around Oven Frame is Attached | F          | 7          | E                             |
| 7               | Oven Frame is put on oven die                | G             | 0.5        | F                             |
| 8               | The Adjustable Base and the two Side are Attached to Oven Frame | H         | 5          | G                             |
| 9               | The Front Panel and Electrical System are Placed and Attached | I       | 2          | H                             |
| 10              | Ignition System is Attached                  | J             | 3          | I                             |
| 11              | Oven Burners are Attached                    | K             | 1.5        | J                             |
| 12              | The Lower Oven Door is Assembly and Attached | L             | 2.5        | K                             |
| 13              | Gas Pipes are assembly and Attached          | M             | 5          | K                             |
| 14              | The Gas is Preliminary Inspected             | N             | 5          | M, L                         |
| 15              | Gas nozzle and Upper Panel are Attached       | O             | 6          | N                             |
| 16              | Gas is Finally Inspected                     | P             | 5          | O                             |
| 17              | Electrical Inspection                        | Q             | 5          | P                             |
| 18              | Back is Assembly and Attached                | R             | 4          | Q                             |
| 19              | Class Cover is Assembly                      | S             | 5          | R                             |
| 20              | The Double Class Oven Door is Assembly       | T             | 6          | R                             |
| 21              | Class Cover is Attached                      | U             | 4          | S                             |
| 22              | Oven Door is Attached                        | V             | 4          | T                             |
| 23              | Support the Accessories                      | W             | 2          | V, U                         |
| 24              | Packaging                                    | X             | 4          | W                             |

4. Practical Application

To evaluate the proposed method, a real life case study of gas cookers assembly line at Light Industrial Company (LIC) which Batool Ibrahim [23] introduced in her work is taken. LIC is a public company in Iraq focuses on durables and apparel sector. For example, different models of gas cookers are produced in the company. The selected gas cooker has the following production specification.
production rate of the company is 35 unit/day, the production line operating time runs 7 hours per day, and the total number of activities to assembly the gas cooker are 24. The sequence of the assembling activities associated with the activity number, name of the activity and the time required to perform the activity are illustrated in Table (2). The precedence diagram of the assembling activities is given in Figure 1. Batool Ibrahim [23] applied the Shortest Operation Time (SOT) for assembly of gas cookers production line.

Figure 1. Precedence diagram of the assembly line of gas cooker [23]

5. Results
In this paper a greedy heuristic algorithm is used for solving SALB-E problem based on a real life case study taken from a literature. The case study is a gas cooker assembly line that is manufactured at the Light Industrial Company in Iraq. The proposed greedy algorithm is described on Table 1. Matlab software package was used to perform the algorithm. Figure 3 show the work element assignment based on the proposed greedy algorithm method. The result obtained from the greedy algorithm is also reported in the last column in Table 3. It can be seen that the number of workstation is 9. The cycle time calculated based on equation (2) is 10. The production rates is 38 unit/day. The Idle time calculated based on equation (3) is 14.5 min/cycle. The efficiency calculated based on equation (4) is 85.35%. In order to evaluate the proposed algorithm, next section compare the result with the performance of the company without applying any assembly line balancing method. The result of this paper is also compared with the results obtained from a previous work in the literature which is based on Shortest Operation Time (SOT) using the software named Quantitative Methods, Production and Operations Management (POM-QM).

6. Discussion
Based on Table 3, the first column presents the performance index of the assemble line balance. The second column presents the actual state of the company, where the assembly line is working without any consideration of workstations. The third column presents the results obtained from a previous work by using POM-QM software and based on SOT. The last column presents the results obtained by using the proposed greedy heuristic algorithm. It can be seen from Table 3 that the greedy heuristic method is more efficient, where the efficiency increased from 78.24% before applying assembly line balancing and 81.64% for SOT method to 85.53% for the greedy heuristic method. In addition, the cycle time based on SOT method is 11.5 min, whereas in the proposed greedy heuristic algorithm is 11min.

| Performance | Company Production Line (actual state) | POM-QM (SOT) | Greedy Heuristics (proposed algorithm) |
|-------------|---------------------------------------|--------------|---------------------------------------|
| No. Workstations | 24 | 9 | 9 |
| Production rates | 35 unit/day | 37 unit/day | 38 unit/day |
| Idle time | 23.5 min/cycle | 19 min/cycle | 14.5 min/cycle |
| Efficiency | 78.24% | 81.64% | 85.35% |

Beside, Figures 2 and 3 show the difference between elements in the workstation based on SOT method and elements in the workstation based on the proposed greedy heuristics algorithm. It can be seen that, in the SOT method workstation 3 contains of 4 elements whereas in the proposed greedy heuristics it contains 3 elements. In the same way, the elements in workstation 4 is 2 based on SOT
method, whereas the elements in workstation 4 is 3 based on the proposed greedy heuristics. The rebalancing the cooker production line as resulted from greedy algorism reduce the lost time between the stations. The application of the greedy algorithm will also reduce the work-in-process items, thus lead to reduce the space required between the workstations.

![Work element assignment based on SOT](image)

**Figure 2.** Work element assignment based on SOT [23]

![Work element assignment based on the proposed greedy algorithm method](image)

**Figure 3.** Work element assignment based on the proposed greedy algorithm method

7. Conclusion

Assembly lines are of great importance in the industrial production of high and low quantity of customized products. The aim of this research is to utilise a greedy heuristic method to solve a single assembly line balancing problem with the objective to enhance the efficiency and effectiveness of production flow. The greedy heuristic method rearranges tasks in stations to minimize idle time. The results obtained from greedy heuristic method were compared with the actual state of the company and the results obtained from the POM-QM Software which is based on SOT. The outcomes of the research have been shown that the greedy heuristic method are more efficient, where the efficiency increased from 78.24% before applying assembly line balancing and 81.64% for SOT method to 85.53% for the greedy heuristic method. The research has recommended that firm should follow the most appropriate method to achieve high efficiency operations.

Reference

[1] Ze-qiang Z, Wen-ming C, Lian-sheng T, and Bin Z 2007 *Int. Conf. on Manag. Sci. and Eng.* IEEE p 369-374.
[2] Roshani A and Giglio D 2015 *IFAC-PapersOnLine*, 48(3) p2299-2304.
[3] Cerqueus A and Delorme X 2019 *Int. J. of P. R.* 57(18) p5640-5659.
[4] Bautista J and Pereira J 2002 *Int. Workshop on Ant Algorithms* (Berlin, Heidelberg, Springer) p 65-75.
[5] Rashid M, Hutabarat W and Tiwari A, 2011 A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches. *The Inter. J. of Adv. Manuf. Techno.* 59(1-4) p335-349.
[6] Seçme G and Özbakır L 2019 *IJORIS* 10(3) p44-58.
[7] Hazır Ö, Delorme X and Dolgui A 2015 A review of cost and profit oriented line design and balancing problems and solution approaches *Annual Reviews in Control* 40 p14-24.
[8] Baybars I 1986 A survey of exact algorithms for the simple assembly line balancing problem *J.Manag. sci.* 32(8) p909-932.
[9] Boysen N, Fleischner M and Scholl A 2008 Assembly line balancing: Which model to use when? *Inter. J. of Prod. Economics* 111(2) p509-528.
[10] Salveson M 1955 The Assembly Line Balancing Problem *J. of Indus. Engin.* 6 p62-69.
[11] Jackson J 1956 A computing procedure for a line balancing problem *J. Manag. Sci.* 2(3) p261-271.
[12] Johnson R 1988 Optimally balancing large assembly lines with (FABLE) *J. Manag. Sci.* 34(2) p240–253.
[13] Hoffmann T 1992 EUREKA: A hybrid system for assembly line balancing *J. Manag. Sci.* 38(1) p39-47.
[14] Scholl A and Klein R 1997 SALOME: A bidirectional branch-and-bound procedure for assembly line balancing *INFORMS J. on Comp.* 9(4) p319-334.
[15] Helgeson W and Birnie D 1961 Assembly line balancing using the ranked positional weight technique *J. of Indust. Engin.* 12(6) p94-398.
[16] Kilbridge M and Wester L 1961 A heuristic method of assembly line balancing *J. of Indust. Engin.* 12(4) p292-298.
[17] Agrawal P 1985 The related activity concept in assembly line balancing *Intern. J. of Prod. Research* 23(2) p403-421.
[18] Suresh G and Sahu S 1994 Stochastic assembly line balancing using simulated annealing *The Intern.J. of Prod. Research* 32(8) p1801-1810.
[19] Chiang W 1998 The application of a tabu search metaheuristic to the assembly line balancing problem *Annals of Operations Research* 77 p209-227.
[20] Anderson E and Ferris M 1994 Genetic algorithms for combinatorial optimization: the assemble line balancing problem *ORSA J. on Computing* 6(2) p161-173.
[21] Lapierre S, Ruiz A and Soriano P 2006 Balancing assembly lines with tabu search *Europ. J.of Operat. Research* 168(3) p826-837.
[22] Scholl A and Becker C 2006 *EJOR* 168(3), pp.666-693.
[23] Ibrahim B 2013 Production flow in gas cooker assembly line *J. of Univ. of Babylon* 21(5) p1752-1761.