Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Enhancing efficiency and adaptability in mixed model line balancing through the fusion of learning effects and worker prerequisites

Esam Alhomaidhi[a*]

[a]*King Fahd University of Petroleum and Minerals, Department of Industrial and Systems Engineering, Dhahran, 31261, Saudi Arabia*

C H R O N I C L E

Article history:
Received July 20 2023
Received in Revised Format October 12 2023
Accepted December 26 2023
Available online December 26 2023

Keywords:
Mixed-model Line balancing
Learning effect
Heuristic
Task requirements
Cost optimization

A B S T R A C T

This research introduces a comprehensive scheme to tackle the Mixed-Model Assembly Line Balancing Problem (MALBPLW) within manufacturing contexts. The primary aim is to optimize assembly line task assignments by integrating both the learning effect and worker prerequisites. The learning effect recognizes the enhanced efficiency of workers over time due to learning and experience. A novel mathematical model and solution approach are proposed, encompassing factors like cycle time, task interdependencies, worker classifications, and the learning effect. The model endeavors to minimize the overall costs related to both workers and workstations while simultaneously maximizing production efficiency. Experimental assessments are conducted to evaluate the efficacy of this proposed approach. Diverse manufacturing scenarios are inspected, comparing and analyzing cost reductions and production efficiency. The outcomes highlight the effectiveness of this approach in achieving enhanced cost-effectiveness and resource utilization in contrast to conventional methods. This study contributes significantly to advancing assembly line balancing and production planning techniques by presenting a pragmatic framework for optimizing resource usage and reducing costs in manufacturing environments. The knowledge extracted from these discoveries can significantly assist professionals in the industry seeking to improve manufacturing processes and strengthen competitiveness.

1. Introduction

In the realm of modern manufacturing, assembly line balancing plays a pivotal role in optimizing efficiency, productivity, and customer satisfaction. With the increasing complexity of production systems, the challenges associated with achieving an ideal assembly line balance are multifaceted. This research paper aims to address the assembly line balancing problem by incorporating crucial factors such as learning effects by task, worker requirements, demand variations, and the complexities of mixed model production lines. Assembly line balancing involves the allocation of tasks and resources across workstations to achieve a harmonious workflow, minimizing idle time, and maximizing output. Traditionally, assembly line balancing has focused on evenly distributing tasks without considering the inherent learning effects associated with each task. However, recognizing that workers become more proficient and efficient with repeated task performance, incorporating learning effects into the line balancing process becomes imperative for achieving optimal results. Worker requirements pose another significant challenge in assembly line balancing. Different tasks may demand varying levels of skills, qualifications, or physical capabilities from workers. Failing to consider these requirements can lead to productivity losses, increased error rates, and worker dissatisfaction. Therefore, an effective assembly line balancing approach should consider matching workers with tasks based on their competencies, enabling smoother operations and improved overall performance. Furthermore, assembly lines frequently encounter demand variations, requiring manufacturers to adapt quickly to changing market needs. Fluctuations in product demand necessitate a flexible production system capable of handling different product models, often

* Corresponding author Tel.:: +966505537398
E-mail esamalhomaidi@kfupm.edu.sa (E. Alhomaidhi)
ISSN 1923-2934 (Online) - ISSN 1923-2926 (Print)
2024 Growing Science Ltd.
doi: 10.5267/j.ijiec.2023.12.008

referred to as mixed model lines. Balancing the workload and ensuring a seamless transition between different product variants on the assembly line is critical for meeting customer demands efficiently and effectively.

This research paper aims to explore and address the challenges associated with assembly line balancing by integrating learning effects, worker requirements, demand variations, and mixed model line complexities. By incorporating learning effects, the proposed methodology accounts for the improving performance of workers over time, leading to increased efficiency and reduced cycle times.

Consideration of worker requirements allows for the assignment of suitable workers to tasks based on their skill sets and capabilities, ensuring a harmonious workflow and mitigating potential bottlenecks. The inclusion of demand variations in the assembly line balancing process enables manufacturers to respond dynamically to changing market demands, providing the necessary flexibility to adjust production rates and adapt to diverse product models. To achieve these objectives, this research paper will explore existing literature and empirical data to develop a comprehensive methodology that integrates learning effects, worker requirements, demand variations, and mixed model line complexities into the assembly line balancing problem. Through the analysis of real-world scenarios and the evaluation of the proposed methodology, this study aims to enhance the understanding of assembly line balancing practices and contribute to the advancement of manufacturing systems.

The subsequent sections of the paper are organized in the following manner: Section 2 presents a comprehensive survey of the literature. Section 3 represents the learning phenomena effect Section 4 introduces the problem definition and the mathematical formulation of the MALBPLW. In Section 5, a solution procedure is developed and presented. The computational performance of the solution procedure is analyzed and compared in Section 6 and 7. Lastly, Section 8 concludes the paper by discussing future research directions.

2. Literature review

The workload distribution problem in the context of assembly line design is commonly known as the assembly line balancing problem (ALBP). The ALBP aims to optimize the assignment of tasks to workstations in an assembly line, considering factors such as task dependencies, cycle time, and resource allocation. The first mathematical formulation for the ALBP was introduced by Salveson (1955) and is known as the Simple Assembly Line Balancing Problem (SALBP). However, SALBP makes simplifying assumptions that limit its applicability to real-world industrial scenarios, Sternatz (2014). As a result, researchers have developed various extensions and enhancements to address more realistic and complex assembly line balancing problems. These extensions consider different aspects such as alternative line layouts (e.g., U-shaped or two-sided lines), diverse product mixes, varying task processing times, and resource allocation requirements. By incorporating these additional factors, researchers aim to develop models and algorithms that better represent real-world assembly line scenarios and provide more effective solutions (Battaïa & Dolgui, 2013; Battaïa & Dolgui, 2022; Boysen et al., 2022; Hazır et al., 2015; Boysen et al., 2007). Within the domain of assembly line balancing, the contributions made can be categorized based on the specific objectives they aim to achieve. These objectives typically revolve around minimizing the number of workstations, reducing cycle time, minimizing costs, or maximizing profit, (Boysen et al., 2008; Hazır et al., 2015). Thomopoulos (1970) initially introduced the mixed-model ALBP, which has since been extensively explored in various avenues. In a subsequent study, Gökċen and Erel (1998) enhanced the mathematical model by incorporating a shortest path reformulation. They further extracted specific characteristics of the MABLP to confine the solution space and devised a heuristic procedure alongside the mathematical model.

Over time, the primary research in this field has expanded to incorporate various line structures, multiple objectives, and advanced algorithms. Vilarinho and Simaria (2002) introduced parallel workstations to the MALBP, which enhances the efficiency of assembly lines by enabling the simultaneous operation of the same sets of tasks at multiple stations. Chutima and Chimklai (2012) tackled a complex multi-objective two-sided MALBP with three objectives. The first objective aimed to minimize the number of Interchangeable stations, while the second objective focused on reducing the number of stations. The third objective encompassed two sub-objectives: work relatedness and workload smoothness. To obtain approximate Pareto solutions, the authors developed a particle swarm optimization algorithm. Delice et al. (2017) integrated a selection mechanism and a novel decoding procedure into a particle swarm optimization algorithm. Their findings demonstrated that this new approach could generate more distinct solutions when compared to the conventional particle swarm optimization algorithm. The research emphasis has shifted towards exploring innovative characteristics and objectives for the MALBP, driven by real-world problems. Tiacci and Mimmi (2018) employed the OCRA index, which measures occupational repetitive action, to evaluate the ergonomic risks faced by workers. They introduced the concepts of blocking and starvation phenomena to model the ergonomic risks associated with workers. The authors' design methodologies aimed to strike a balance between ergonomic benefits and production profits. In a different study, Sun and Fan (2018) introduced the notion of variety-induced changeover complexity in mixed-model assembly lines. They proposed entropy-based methods to measure three types of changeover complexities. To address a car sequencing problem with the objective of minimizing changeover complexities, they employed a multi-objective ant colony algorithm. Chen et al. (2019) focused on a MALBP in the TFT–LCD module process, taking into account practical characteristics such as multi-skilled workers and operator efficiency. They developed a heuristic two-phase adaptive genetic algorithm (AGA) to tackle the problem. Considering energy consumption, Zhang et al.

(2020) incorporated an energy objective into the MALBP. The authors identified that idle state energy consumption constituted a significant portion of the total consumption. Battaïa et al. (2015) tackled an MALBP that took into account the assignment of operations and disabled workers, who are multi-skilled workers. They presented a formulation using mixed-integer linear programming and suggested constructive heuristics as solutions for the problem.

Delorme et al. (2019) conducted a study on a paced assembly line planning problem with the objective of minimizing the number of identical workers. The researchers considered multiple types of operations, as well as lower and upper bounds for the required workforce and a predetermined cycle time. To address this problem, they developed a mixed-integer linear programming model, an enumeration algorithm, and a dynamic programming algorithm. The ALBP literature has increasingly focused on the practical relevance of the task learning effect, Glock et al. (2019). Empirical studies have demonstrated that task times decrease as the number of task repetitions increases (Otto & Otto, 2014; Alhomaidi & Askin, 2022). This occurrence has attracted considerable focus within the field.

Cohen et al. (2006) tackled a work allocation problem with the objective of minimizing the makespan of production, assuming homogeneous rates. They highlighted that considering task learning can lead to significant reductions in makespan, particularly in scenarios with low overall demand. Toksarı et al. (2008) developed a procedure for obtaining polynomial solutions for the ALBP with incorporated task learning effects. The authors established that the optimal solution must adhere to the shortest task time rule. Addressing an automated flexible ALBP with collaborative task learning, Li and Boucher (2017) introduced a task reassignment procedure and backward induction rules. Their approach aimed to achieve optimal efficiency in realtime production. In the study by Li (2017), a stochastic learning curve was proposed to account for the occasional and inconsistent magnitude of task time improvement. Building upon this observation, the stochastic learning effect was incorporated into the conventional ALBP model, and an algorithm called ENCORE was developed to solve it. To validate the efficiency of the proposed algorithm, statistical experiments were conducted. Examining the impact of learning on the cycle time in a simple assembly line, Koltai and Kalló (2017) conducted an analysis. They also performed a sensitivity analysis regarding learning rates and discovered that the number of bottleneck shifts increases as the learning rate decreases.

Some key research works mentioned above have been summarized in Table 1. Previous studies have explored MALBP with various additional features. In practical applications, MALBP often involves mass customization processes characterized by low overall production volume and short makespan for each product model. It is widely recognized that task time significantly decreases during the initial phase of production due to learning, reaching a plateau as the production volume increases. Therefore, considering the learning effect can greatly enhance production efficiency, particularly in mass customization scenarios with low overall production volume, compared to mass production scenarios with high overall production volume. This paper incorporates the learning effect, worker categories, and decision-making regarding production volume into a comprehensive MABLP.

Table 1:
Literature summary

Term	Definition/Description	Features/Objectives
Thomopoulos (1970)	Customized heuristic algorithm	Initial study MALBP
Dolgui et al. (2018)	Conventional and randomized heuristics	Minimize the number of workers
Delorme et al. (2019)	Enumeration algorithm and a dynamic programming	Minimize the cycle time
Battaïa et al. (2015)	Conventional and randomized heuristics	Minimize the total number of workers
Vilarinho and Simaria (2002)	Two-stage simulated annealing	Parallel workstations, MALBP
Chutima and Chimklai (2012)	Particle swarm optimization	Two-sided MALBP
Gökc˙en and Erel (1998)	Binary integer programming	MALBP
Sun and Fan (2018)	Ant colony optimization	Changeover complexity, MALBP
Chen et al. (2019)	Adaptive genetic algorithm	Multi-skilled workers, MALBP
Zhang et al. (2020)	Cellular genetic algorithm	Energy consumption and sequencing, MALBP
Li and Boucher (2017)	Heuristic algorithm	Stochastic learning curve
Toksarı et al. (2008)	Shortest path time rule	Learning effect, U-shaped line
Koltai and Kalló (2017)	Theoretical analysis	Learning effect
Otto and Otto (2014)	Priority rules-based method	Learning effect
Alhomaidi and Askin (2022)	Practical analysis and mixed integer programming	Learning effect and periodical demand
Cohen et al. (2006)	Theoretical analysis	Learning effect and minimize makespan
This paper	Integer programing and heuristic algorithm	Minimize the number of workstations, workers and total cost

3. Task's Learning Effect

The proposed model builds upon the widely-explored SALBP (Sequential Assembly Line Balancing Problem) introduced by Scholl in Chapter 2 of their work Scholl (1999). The SALBP serves as the fundamental basis for the proposed model and has been extensively studied in various research studies related to assembly line balancing problems, as referenced in Boysen et al. (2008) and Becker and Scholl (2006). In this model, the production process involves the assembly of a homogeneous product using *k* similar workstations. The task assignments to the workstations must satisfy pre-specified precedence constraints, ensuring the correct order of tasks where the production is carried out on a serial, paced assembly line. In order to incorporate the concept of learning into the model, the Yelle learning curve, as presented by Yelle (1979), is employed. This learning curve builds upon the research outlined in Wright (1936) and offers a structured approach to capture the impact of learning in production processes. The learning curve is applied to each specific repetition Θ of task *i*, and its mathematical representation is as follows:

$$\mathrm{p_i}(\Theta) = \mathrm{p_i^1} \cdot (\Theta)^{\frac{\log(\mathrm{r_i})}{\log(2)}} \tag{1}$$

In Eq. (1), $p_i(\Theta)$ represents the required amount of time to process task *i* during repetition Θ. $p_i(1)$ denotes the processing time required to complete the first unit of task *i*. r_i represents the learning rate for task *i*. The learning curve assumes that the processing time for a task decreases monotonically as the number of repetitions increases, indicating the accumulation of experience. In other words, for each task *i*, the processing time for repetition Θ + 1 is less than or equal to the processing time for repetition Θ, as stated in the inequality $p_i(\Theta + 1) \leq p_i(\Theta)$. This assumption holds for all tasks *i*, ($i = 1,...,N$) and all repetitions Θ, ($\Theta = 1,...,R$). Moreover, the proposed model considers the variability of learning rates among different tasks, which can be influenced by factors such as task complexity, required dexterity, or the level of automation. This consideration provides the model with the flexibility to capture the diverse learning effects across tasks accurately. In terms of the temporal aspect, the model divides time into discrete periods. It also accommodates the possibility of a ramp-up phase or other variable factors affecting production volume. Consequently, the demand for the product may fluctuate across these periods throughout the planning horizon. It is important to emphasize that while demand may vary, the assignments of tasks to workstations remain consistent across all cycles. This consistency ensures that workers can specialize and develop specific skills related to their assigned tasks.

4. Problem definition

The problem being addressed in this study is a new extension of the work related to the mixed model assembly line balancing problem with the incorporation of the learning effect and consideration of worker-type requirements for each task. In a manufacturing setting, multiple product models are produced on an assembly line with a series of workstations. The goal is to assign tasks to workstations optimally, considering various factors such as cycle time, task dependencies, worker categories, and the learning effect. The learning effect applies to how workers progressively improve their task efficiency through experience and learning over time. As workers repeat a task, their processing time decreases, resulting in increased productivity. The problem also considers the worker-type requirements for each task. Different tasks may require specific worker capabilities or skills, and it is important to assign tasks to workers who possess the necessary qualifications.

To address the problem efficiently, a novel mathematical model and a solution method are proposed, which aim at contributing to the development of a new framework for production planning, incorporating the learning effect in task assignments and worker categories, with the goal of minimizing the total cost associated with the total number of workers and workstations. The presented problem and proposed methods encompass the following aspects:

- Task Learning Effect: The model considers the learning effect associated with each task, capturing the relationship between task time reductions and increased worker experience. By incorporating the learning effect, the framework optimizes task assignments, harnessing the improved efficiency gained through worker experience.

- Worker Categories: The decision-making process involves determining the worker categories required for each task. This ensures that tasks are assigned to workers with the appropriate skills and expertise, facilitating an efficient production process.

- Minimization of Total Cost: The proposed framework addresses the objective of minimizing the total cost associated with the total number of workers and workstations required for production. By optimizing task assignments based on the learning effect and worker categories, the framework aims to achieve an efficient allocation of resources, reducing overall costs and improving cost effectiveness.

- High-Quality Solutions: The developed algorithm is designed to deliver high quality solutions. Even under real-time conditions, the algorithm generates effective production plans that optimize resource utilization and minimize costs.

By incorporating the learning effect, worker categories, and the objective of minimizing the total cost associated with workers and workstations, the proposed framework offers an innovative approach to production planning. It optimizes task assignments based on worker experience, assigns tasks to appropriate worker categories, and reduces overall costs, resulting in improved cost-effectiveness. The algorithms provide practical solutions that can be implemented in real-time scenarios, enabling manufacturers to optimize their production processes while minimizing costs associated with workers and workstations. The problem researched in this paper is denoted as the Mixed-Model Assembly Line Balancing Problem by Incorporating Task Learning Effect and Worker Requirements (MALBPLW).

4.1. Exact Model

Sets and Parameters

i	Task number, i=1,2, … , N
ψ	Period index, ψ = 1, 2, . . . ,R
m	Model number, m = 1, 2, . . . ,M
k	Workstation number, k=1,2, … , K
e	Worker category, e=1,2, … , E
D_m	Demand of model m
d_ψ	Cumulative demand for all models for period ψ.
IP_m	Ordering set (g,i) of tasks such that task g must precede task i for model m (needed only for immediate predecessors).
WS	Predetermined depreciation cost for workstation k.
WT_e	Hiring cost for a worker with category e.
ζ_{ie}	Binary indicator equal to 1 if task i requires worker category e and 0 otherwise.
λ	Maximum number of workers assigned within a workstation.
Θ	Recurrence number.
r_{im}	Learning rate for task i belongs to model m.
p_{im}^1	Processing time of task i for the first unit belongs to model m.

Decision variable:

$$x_{imk} = \begin{cases} 1, & if\ task\ i\ for\ model\ m\ is\ assigned\ to\ workstaion\ k \\ 0, & Otherwise \end{cases} \quad \forall i \in N, \forall m \in M, \forall k \in K \tag{2}$$

$$S_k = \begin{cases} 1 & if\ workstation\ k\ is\ constructed\ within\ the\ line \\ 0, & Otherwise \end{cases} \quad \forall k \in K \tag{3}$$

$$w_{ek} = \begin{cases} 1, & if\ worker\ with\ category\ e\ is\ assigned\ to\ workstaion\ k \\ 0, & Otherwise \end{cases} \quad \forall e \in E, \forall k \in K \tag{4}$$

c_Θ: Cycle time during recurrence Θ (5)

Objective:

$$\min \text{Cost} = \sum_{k=1}^{K} WS * s_k + \sum_{k=1}^{K} \sum_{e=1}^{E} WT_e * w_{ek} \tag{6}$$

Constraints:

$$\sum_{k=1}^{K} x_{imk} = 1 \qquad \forall i, \forall m \tag{7}$$

$$x_{imq} - \sum_{k=1}^{q} x_{gmk} \leq 0 \qquad \forall q, (g,i) \in IP_m, \forall m \tag{8}$$

$$\sum_{m=1}^{M} \sum_{i=1}^{N} \left(p_{im}(1) * \Theta^{\frac{log_2}{log(r_{im})}} \right) * x_{imk} \leq c_\Theta \qquad \forall k,\ \forall \Theta \tag{9}$$

$$\sum_{\Theta=d_\psi+1}^{d_{\psi+1}} c_\Theta \leq T \qquad \forall \psi \tag{10}$$

$$\zeta_{ie} * x_{imk} \leq w_{ek} \qquad \forall i, \forall m, \forall k,\ \forall e \tag{11}$$

$$\sum_{e=1}^{E} w_{ek} \leq \lambda \qquad \forall k \tag{12}$$

$$x_{imk} \leq s_k \qquad \forall i,\ \forall m,\ \forall k \tag{13}$$

$$x_{imk} \in \{0,1\} \qquad \forall i,\ \forall m,\ \forall k \tag{14}$$

$$s_k \in \{0,1\} \qquad \forall k \tag{15}$$

$$w_{ek} \in \{0,1\} \qquad \forall e,\ \forall k \tag{16}$$

$$c_\Theta \geq 0 \qquad \forall \Theta \tag{17}$$

The multi-objective function (6) is to minimize the total cost associated with the total number of workstations and worker categories simultaneously. Constraint (7) guarantees that each task is allocated to one workstation only. Constraint (8) enforces the precedence constraints specified by the ordering set IP_m. Constraint (9) limits the total processing time of tasks assigned to each workstation during the learning occurrence. This prevents any cycle time violation during any recurrence Θ. Constraint (10) ensures that the cumulative cycle time does not exceed the available time. Constraint (11) guarantees that if a task i requires a worker category e, then the worker must be assigned to the workstation where the task is assigned. In the case of limited space, Constraint (12) limits the number of workers assigned to each workstation to be at most λ workers. Constraint (13) ensures that a task is assigned to a workstation only if the workstation is constructed. Constraints (14), (15), and (16) define the binary nature of the decision variables.

5. MALBPLW Solution Procedure

The assembly line balancing problem poses a challenge in terms of its combinatorial complexity, making it difficult for traditional mathematical procedures or exact solution methods to handle large-scale instances of the problem (i.e., NP-Hard, Wee and Magazine (1982). As a solution, an algorithmic approach in the form of a heuristic is developed to provide a fast and efficient method for finding solutions. This heuristic algorithm aims to effectively tackle the problem and deliver satisfactory results within a reasonable timeframe. In order to provide a clearer understanding of the heuristic, it is beneficial to introduce and discuss several key concepts beforehand. This will simplify the establishment of a strong foundation and enable a more seamless presentation of the heuristic approach. In mixed-model assembly lines, multiple models often involve common tasks with similar precedence relationships. Taking advantage of this similarity, the model incorporates the shared precedence relations between different models. This is achieved by adopting a combined precedence diagram, as proposed by Thomopoulos Thomopoulos (1970), which connects the precedence relationships of multiple models on a single diagram. By utilizing this combined diagram, the model can effectively capture and represent the precedence connections across various models. Furthermore, considering that the demand for each model may differ, the utilization of workstations can be measured by using the weighted average task duration for each task i, This approach takes into account the varying demand by assigning weights corresponding to each model, resulting in a more precise assessment of workstation utilization.

5.1. Heuristic Description

At the onset of the algorithm, various notations and variables are introduced to establish a consistent framework for the subsequent steps. These defined notations and variables are utilized throughout the algorithm to ensure clear and consistent representation of the problem and its solution. Notation:

p_{im}	Weighted average task time for task i. Model type, $m = 1,2,...,M$,
c	Cycle time,
L	List of unassigned tasks,
e_i	Worker category required for task I,
d_m	Demand model m,
d	Cumulative demand for all models,
W_{ek}	Worker categories set assigned to workstation k,
f_k	Remaining available time per cycle for each workstation k,
Q	Prioritized tasks list,
Θ_m	Recurrence number for each model,
r_{im}	Learning rate for task i belongs to model m,
p_{im}^1	Processing time of task i for the first unit belongs to model m,
WT_e	Hiring cost for a worker with category e,
WS	Predetermined depreciation cost for workstation k.

Algorithm 1 MALBPLW Algorithm

```
1: Procedure Construction ($Q$, $p_{im}$, $\Theta_m$, $r_{im}$, $d_m$, $c$, $e_i$)
2:   for each model's individual task do
3:     Calculate the learning curve: $p_{im}(1) * \Theta^{\frac{log_2}{log(r_{im})}}$ where $\Theta_m = d_m$
4:   end for
5:   for each task $i$ do
6:     Compute the weighted average task time
7:   end for
8:   Prioritize the tasks based on the positional weight $Q$
9:   Set $W_{ek} = \emptyset$, $f_k = c_h$, and $k = 1$
10:  while $Q \neq \emptyset$ do
11:    Select next $q_i$ from $Q$ and update $Q = Q - \{q_i\}$
12:    Assign task $q_i$:
13:    if $p_{q(i)} \leq f_k$ and all predecessors are met then
14:      Assign $q_i$ to the current workstation
15:      if required worker $e_i$ is not already assigned then
16:        Assign worker category $e$
17:      end if
18:      Update $W_k = W_k \cup e_{(i)}$, $f_k = f_k - p_q(i)$, and $i = i + 1$
19:    else
20:      Find the next first fitable task $q_f$ in $Q$
21:      Exchange $q_i$ with $q_f$ in $Q$
22:    end if
23:    if time capacity violation in current workstation then
24:      Find a task switching pair $(i,h)$ in $U$
25:      if $p_h - p_i \leq f_k$ and $e_i$ requirements are satisfied then
26:        Switch $i$ with $h$
27:        Update $Q$, $f_k$, and $U = U - \{(i,h)\}$
28:      else
29:        Create a new workstation
30:        Set $k = k + 1$ and $f_k = c$
31:      end if
32:    end if
33:  end while
34:    Calculate the Total construction cost: $WT_e * W_{ek} + k * WS$
35: end procedure
```

The following steps outline the general process of assigning tasks and workers to workstations at a high level.

- For each individual task of each model, the algorithm calculates the learning curve using the formula ($p_{im}(1) * \Theta_m^{\frac{log_2}{log(r_{im})}}$), where $\Theta_m = d_m$. This step helps estimate the processing time for each task based on the learning rate and recurrence number.
- The algorithm computes the weighted average task time for each task i. This is done to determine the average time required to complete each task, considering the variability in task duration across different models.
- The algorithm prioritizes the tasks based on their positional weight. The prioritized task list is denoted as Q.
- The algorithm initializes the variables W_{ek} (the set of worker categories assigned to each workstation), f_k (the remaining available time per cycle for each workstation k), and k (the workstation index).
- The algorithm iterates in a loop until the task list Q becomes empty.

- In each iteration of the loop, the algorithm selects the next task q_i from the list Q and removes it from the list.

- The task q_i is assigned to a workstation based on certain conditions:

- If the processing time $p_{q(i)}$ of task q_i is less than or equal to the remaining available time f_k of the current workstation and all its predecessors are met, the task is assigned to the workstation. If the required worker category e_i is not already assigned to the workstation, it is assigned. The variables W_k, f_k, and i are updated accordingly.

- If the time capacity of the current workstation is violated (i.e., the processing time of the task exceeds the remaining available time), the algorithm searches for the next first fitable task q_f in the list Q. The task q_i is then exchanged with q_f in the list Q to ensure the precedence and worker category requirements are met.

- If the current workstation cannot accommodate any additional tasks from Q due to time capacity violation, the algorithm enters a nested loop:

- It looks for a task switching pair (i,h) in the task switching set U.

- If there is a pair (i,h) where the time difference $p_h – p_i$ is less than or equal to the remaining available time f_k of the current workstation, and the required worker e_i is either already assigned or will not be required again, and all predecessors of h are met, the algorithm switches task i with h. The task switching set U, the remaining available time f_k, and the task list Q are updated accordingly.

- If no suitable task switching pair is found, the algorithm creates a new workstation by incrementing the workstation index k, sets the remaining available time f_k to the cycle time c, and goes back to the original list.

- Once all the tasks are assigned to workstations, the algorithm calculates the total construction cost by multiplying the hiring cost WT_e with the set of assigned worker categories W_{ek}, and adding the product to the product of the workstation count k and the predetermined depreciation cost per workstation WS.

6. Computational Experiment

The proposed method's efficiency is evaluated using a well-known dataset referenced as Gökçen et al. (2006) in the literature. The experiment encompasses different instance sizes, namely small, medium, and large, including the datasets named $Jackson - 11$, $LUTZ - 32$, $Tong - 68$, $WEEMAG - 75$, and $ARC - 111$. Each dataset is subjected to various cycle times. In this experiment, three models with diverse demands are considered. The learning rate for each task is randomly assigned from a uniform distribution, $U \sim [85\%, 100\%]$. Additionally, four categories of workers are employed, with three being professionals and the remaining one being regular workers who possess minimal skills to process most of the tasks. The task requirements are distributed randomly. The hiring costs for each worker type, workstation installation expenses, and other relevant parameters can be found in Table 2.

Table 2
Parameters values

Parameter	Value
rim	$U\sim [85\%, 100\%]$
Hiring cost	Category A (regular): \$60, category B: \$80; category C: \$160; category D: \$240
WS Cost	\$3000
Number of models	3
Demand	35, 46 and 70

To handle the randomness involved in the algorithm, each test combination is run 10 times with a total of 250 runs. To assess and compare the heuristic for larger problems, it is necessary to first establish a strong lower bound to find the theoretical minimum number of workstations, denoted as K_{min}, needed to fulfill the customer's demand (cycle time constraint). This is in addition to a total cost lower bound that includes labor hiring cost and workstation installation cost. Through computation, the theoretical lower bound for the number of workstations is determined.

$$LB_k = \left\lceil \left(\sum_{m\in M} \sum_{i=1}^{N} \frac{p_{im}}{c} \right) \right\rceil \tag{18}$$

Thus, the lower bound for the total cost, which considers workstation installation costs and labor hiring costs, is determined by the following expression:

$$LB_c = LB_k\left(WS + WT_{typeA}\right) + \sum_{\substack{\forall e \\ e \neq typeA}} WT_e \tag{19}$$

The first term represents the cost associated with the workstation cost plus the loading of one regular worker within each of them. The latter reports one worker only from each category other than the regular worker within the whole line. The presented model is scripted in Python, subsequently executed and resolved on a computer featuring an Intel i7-4790 CPU operating at 3.60GHz and equipped with 16 GB RAM.

7. Computational Results

The experiment utilizes four main criteria to evaluate the effectiveness of the heuristic. These criteria include the count of workstations, the overall cost throughout a specific time, the variation between the heuristic's outcome and the lower bound of total cost, and the execution time of the CPU. Table 3 and Fig. 1 provide an overview of the outcomes achieved by a heuristic algorithm for different instances. It consists of several columns, including Instance information, cycle time value, execution time in seconds, number of workstations, Labor cost, and Total Cost. Each row corresponds to a specific instance and displays the corresponding values for these metrics. By examining the table, certain patterns are observed. For example, as the parameter value "c" increases for a given instance, a decrease in the CPU time taken by the algorithm is obtained. This suggests that higher values of "c" result in more efficient computations. Additionally, an increase in "c" tends to lead to a reduction in the number of workstations required, as indicated by the "K" column. This implies that higher parameter values allow for more effective utilization of existing resources. Furthermore, the result reveals a consistent relationship between "c" and the associated labor and total cost. As the parameter value increases, both labor and total cost tend to decrease. This suggests that higher values of "c" contribute to improved efficiency and cost-effectiveness in terms of labor requirements and overall expenses. To further examine the presented heuristic, a comparison is made between the obtained results and the total cost and theoretical workstation lower bound. As seen from Fig. 2 and Table 4, the total cost shortfalls between both outcomes are in the range of 3% and 9% in most cases, diminished within an average of 6.68% as observed in the last column. Additionally, the heuristic attains the theoretical number of workstations in most cases. Table 5 clearly demonstrates the advantage of incorporating learning in the design, with savings reaching as high as 51.2%. On average, the savings amount to around 32%. This reduction in cost is primarily attributed to the decreased number of workstations required and the resulting labor cost savings, which stem from the acquired skills during the manufacturing process.

Table 3
Heuristic outcomes

Instance	*c*	*CPU Time (sec)*	*K*	*Labor*	*Total Cost*
Jackson11	7	0.470	5	1,180	16,180
	9	0.421	4	1,120	13,120
	10	0.377	4	1,120	13,120
	12	0.338	3	1,060	10,060
	16	0.303	2	840	6,840
LUTZ32	115	0.563	7	2,020	23,020
	129	0.520	6	1,960	19,960
	140	0.480	5	1,840	16,840
	154	0.443	5	1,760	16,760
	175	0.409	4	1,720	13,720
Tong68	110	0.607	11	3,020	36,020
	125	0.579	9	2,960	29,960
	134	0.552	10	2,200	32,200
	150	0.526	8	2,760	26,760
	165	0.502	7	2,640	23,640
WEE-MAG75	121	2.270	11	3,460	36,460
	139	2.142	9	3,140	30,140
	145	2.021	9	3,360	30,360
	153	1.907	8	3,660	27,660
	166	1.800	8	3,160	27,160
ARC111	65	3.969	20	4,160	64,160
	73	3.492	18	4,360	58,360
	80	3.073	16	4,480	52,480
	86	2.705	15	4,360	49,360
	90	2.380	14	4,220	46,220

Table 4
Heuristic performance and efficiency

Name	C	Theoretical K	LB	Dev. %
Jackson11	7	5	15,780	2.53%
	9	4	12,720	3.14%
	10	4	12,720	3.14%
	12	3	9,660	4.14%
	16	2	6,600	3.64%
LUTZ32	115	7	21,900	5.11%
	129	6	18,840	5.94%
	140	5	15,780	6.72%
	154	5	15,780	6.21%
	175	4	12,720	7.86%
Tong68	110	11	34,140	5.51%
	125	9	28,020	6.92%
	134	9	28,020	14.92%
	150	8	24,960	7.21%
	165	7	21,900	7.95%
WEE-MAG75	121	11	34,140	6.80%
	139	9	28,020	7.57%
	145	9	28,020	8.35%
	153	8	24,960	10.82%
	166	8	24,960	8.81%
ARC111	65	19	58,620	9.45%
	73	18	55,560	5.04%
	80	16	49,440	6.15%
	86	15	46,380	6.43%
	90	14	43,320	6.69%

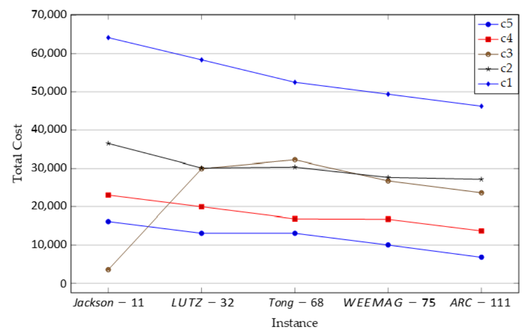

Fig. 1. Heuristic's total cost for different cycle time

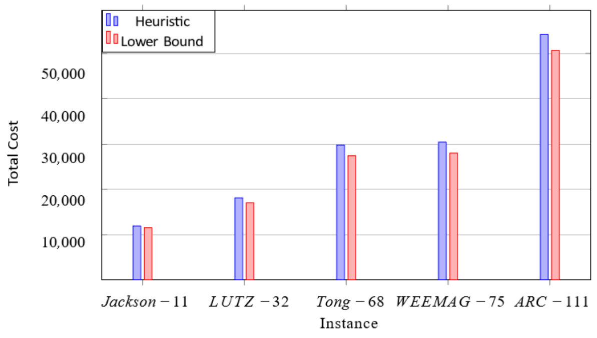

Fig. 2. Heuristic Vs lower bound total cost

Table 5
Learning advantage analysis

Instance	*C*	*K*	*Labor*	*Total Cost*	*Saving dev. %*
Jackson11	7	8	1,040	24,410	50.9%
	9	6	880	18,260	39.2%
	10	6	880	18,260	39.2%
	12	5	880	15,210	51.2%
	16	3	720	9,120	33.3%
LUTZ32	115	9	1,840	27,590	19.9%
	129	9	2,000	27,610	38.3%
	140	8	2,080	24,540	45.7%
	154	7	2,320	21,500	28.3%
	175	6	2,160	18,420	34.3%
Tong68	110	15	4,000	46,550	29.2%
	125	13	3,760	40,250	34.3%
	134	12	3,280	37,070	15.1%
	150	11	3,440	33,980	27.0%
	165	10	3,280	30,860	30.5%
WEE-MAG75	121	15	3,920	46,540	27.6%
	139	13	4,000	40,280	33.6%
	145	13	3,440	40,210	32.4%
	153	12	3,760	37,130	34.2%
	166	11	3,520	33,990	25.1%
ARC111	65	27	5,680	85,220	32.8%
	73	24	5,040	75,390	29.2%
	80	22	5,040	68,940	31.4%
	86	20	4,720	62,490	26.6%
	90	19	4,240	59,240	28.2%

8. Conclusion and Future Research

This study addresses the mixed-model assembly line balancing problem by incorporating the learning effect and considering worker-type requirements for each task. The proposed approach, based on integer programming and heuristic techniques, offers an innovative solution to production planning by optimizing task assignments based on worker experience, assigning tasks to appropriate worker categories, and minimizing overall costs associated with workers and workstations. The study contributes to the development of a new methodology for production planning by formulating the problem as an integer programming model. The model captures the learning effect and worker-type requirements, allowing for the optimization of task assignments. Additionally, a heuristic technique is employed to tackle the computational complexity of the problem and provide efficient solutions. The computational performance of the proposed algorithm is designed to deliver high-quality solutions, even under real-time conditions. It generates effective production plans that optimize resource utilization and minimize costs, providing practical solutions for manufacturers. However, there are several avenues for future research to further advance assembly line balancing in manufacturing. Firstly, investigating advanced machine learning and artificial intelligence techniques may enhance the accuracy of predicting the learning effect and optimizing task assignments. Deep learning models and reinforcement learning algorithms could be explored to handle more complex assembly line scenarios and achieve even better performance. Additionally, extending the research to include other factors such as machine breakdowns, maintenance schedules, and production uncertainties would provide a more comprehensive framework for production planning. By considering these factors, the proposed approach can be made more robust and adaptable to real-world manufacturing settings. Lastly, further empirical studies and collaborations with industry partners are necessary to validate the proposed framework in real-world manufacturing environments. The integration of Industry 4.0 technologies and exploring the implications for the broader supply chain can also provide valuable insights.

Acknowledgement

The author wishes to acknowledge the support of King Fahd University of Petroleum and Minerals.

References

Alhomaidi, E., & Askin, R. G. (2022). The Assembly Line Balancing Problem in the Presence of Task Learning and Demand Fulfilment (ALBLDP). In IIE Annual Conference. *Proceedings (pp. 1-6). Institute of Industrial and Systems Engineers (IISE).*

Battaïa, O., Delorme, X., Dolgui, A., Hagemann, J., Horlemann, A., Kovalev, S., & Malyutin, S. (2015). Workforce minimization for a mixed-model assembly line in the automotive industry. *International Journal of Production Economics, 170, 489-500.*

Battaïa, O., & Dolgui, A. (2013). A taxonomy of line balancing problems and their solution approaches. *International journal of production economics, 142*(2), 259-277.

Battaïa, O., & Dolgui, A. (2022). Hybridizations in line balancing problems: A comprehensive review on new trends and formulations. *International Journal of Production Economics,* 108673.

Boysen, N., Fliedner, M., & Scholl, A. (2007). A classification of assembly line balancing problems. *European journal of operational research, 183*(2), 674-693.

Boysen, N., Fliedner, M., & Scholl, A. (2008). Assembly line balancing: Which model to use when? *International journal of production economics, 111*(2), 509-528.

Boysen, N., Schulze, P., & Scholl, A. (2022). Assembly line balancing: What happened in the last fifteen years? *European Journal of Operational Research, 301*(3), 797-814.

Chen, J. C., Chen, Y. Y., Chen, T. L., & Kuo, Y. H. (2019). Applying two-phase adaptive genetic algorithm to solve multi-model assembly line balancing problems in TFT–LCD module process. *Journal of Manufacturing Systems, 52,* 86-99.

Chutima, P., & Chimklai, P. (2012). Multi-objective two-sided mixed-model assembly line balancing using particle swarm optimization with negative knowledge. *Computers & Industrial Engineering, 62*(1), 39-55.

Cohen, Y., Vitner, G., & Sarin, S. C. (2006). Optimal allocation of work in assembly lines for lots with homogenous learning. *European Journal of Operational Research, 168*(3), 922-931.

Delice, Y., Kızılkaya Aydoğan, E., Özcan, U., & İlkay, M. S. (2017). A modified particle swarm optimization algorithm to mixed-model two-sided assembly line balancing. *Journal of Intelligent Manufacturing, 28,* 23-36.

Delorme, X., Dolgui, A., Kovalev, S., & Kovalyov, M. Y. (2019). Minimizing the number of workers in a paced mixed-model assembly line. *European Journal of Operational Research, 272*(1), 188-194.

Dolgui, A., Kovalev, S., Kovalyov, M. Y., Malyutin, S., & Soukhal, A. (2018). Optimal workforce assignment to operations of a paced assembly line. *European Journal of Operational Research, 264*(1), 200-211.

Glock, C. H., Grosse, E. H., Jaber, M. Y., & Smunt, T. L. (2019). Applications of learning curves in production and operations management: A systematic literature review. *Computers & Industrial Engineering, 131,* 422-441.

Gökçen, H., Ağpak, K., & Benzer, R. (2006). Balancing of parallel assembly lines. *International Journal of Production Economics, 103*(2), 600-609.

Gökċen, H., & Erel, E. (1998). Binary integer formulation for mixed-model assembly line balancing problem. *Computers & industrial engineering, 34*(2), 451-461.

Hazır, Ö., Delorme, X., & Dolgui, A. (2015). A review of cost and profit oriented line design and balancing problems and solution approaches. *Annual Reviews in Control, 40,* 14-24.

Koltai, T., & Kalló, N. (2017). Analysis of the effect of learning on the throughput-time in simple assembly lines. *Computers & industrial engineering, 111*, 507-515.

Li, Y. (2017). The type-II assembly line rebalancing problem considering stochastic task learning. *International Journal of Production Research, 55*(24), 7334-7355.

Li, Y., & Boucher, T. O. (2017). Assembly line balancing problem with task learning and dynamic task reassignment. *The International Journal of Advanced Manufacturing Technology, 88,* 3089-3097.

Otto, C., & Otto, A. (2014). Extending assembly line balancing problem by incorporating learning effects. *International Journal of Production Research, 52*(24), 7193-7208.

Salveson, M. E. (1955). The assembly-line balancing problem. *Transactions of the American Society of Mechanical Engineers, 77*(6), 939-947.

Scholl, A., & Scholl, A. (1999). Balancing and sequencing of assembly lines *(pp. 34-351). Heidelberg: Physica-Verlag.*

Sternatz, J. (2014). Enhanced multi-Hoffmann heuristic for efficiently solving real-world assembly line balancing problems in automotive industry. *European Journal of Operational Research, 235*(3), 740-754.

Sun, H., & Fan, S. (2018). Car sequencing for mixed-model assembly lines with consideration of changeover complexity. *Journal of manufacturing systems, 46,* 93-102.

Thomopoulos, N. T. (1970). Mixed model line balancing with smoothed station assignments. *Management science, 16*(9), 593-603.

Tiacci, L., & Mimmi, M. (2018). Integrating ergonomic risks evaluation through OCRA index and balancing/sequencing decisions for mixed model stochastic asynchronous assembly lines. *Omega, 78,* 112-138.

Toksarı, M. D., İşleyen, S. K., Güner, E., & Baykoç, Ö. F. (2008). Simple and U-type assembly line balancing problems with a learning effect. *Applied Mathematical Modelling, 32*(12), 2954-2961.

Vilarinho, P. M., & Simaria, A. S. (2002). A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations. *International journal of production research, 40*(6), 1405-1420.

Wee, T. S., & Magazine, M. J. (1982). Assembly line balancing as generalized bin packing. *Operations Research Letters, 1(2), 56-58.*

Wright, T. P. (1936). Factors affecting the cost of airplanes. *Journal of the aeronautical sciences, 3*(4), 122-128.

Yelle, L.E., 1979. The learning curve: Historical review and comprehensive survey. *Decision sciences 10,* 302–328.

Zhang, B., Xu, L., & Zhang, J. (2020). A multi-objective cellular genetic algorithm for energy-oriented balancing and sequencing problem of mixed-model assembly line. *Journal of Cleaner Production, 244,* 118845.