

# Analysis of the Effect of Learning on the Solution of a Simple Assembly Line Balancing Problem

Zakaria Zine El Abidine, Imre Dimény and Tamás Koltai

Department of Management and Business Economics

Budapest University of Technology and Economics

H-1521, Budapest, P.O.B. 91, Hungary

[zakariazineelabidine@edu.bme.hu](mailto:zakariazineelabidine@edu.bme.hu), [dimeny.imre@gtk.bme.hu](mailto:dimeny.imre@gtk.bme.hu), [koltai.tamas@gtk.bme.hu](mailto:koltai.tamas@gtk.bme.hu)

## Abstract

The objective of simple assembly line balancing problem type I is to minimize the number of workstations organized to perform tasks with precedence constraints. Assembly lines, which require a lot of manual labor, are an example of a manufacturing environment where learning effect influences the operation. In this case, when the minimal number of workstations is determined, the change of task times as a consequence of learning must be considered. This paper investigates the effects of two exponential learning curves on the minimal number of workstations when a simple assembly line balancing problem is solved. A modified simple assembly line balancing model incorporating the learning effect is formulated, and a sample problem to illustrate the application of the model is provided. Some general conclusions related to the type of learning curve function applied and to the change of learning rates are also presented.

## Keywords:

Assembly line balancing, simple assembly line, learning curve, production management, mathematical programming.

## 1. Introduction

Learning is a process that all humans go through. The well-known exponential learning function (Wright 1936) formalized first this empirical evidence in manufacturing. The early applications studied how the number of units increases or the unit production costs decrease as the cumulative number of manufactured products increases. Since then, learning curves have been extensively used to represent the dynamics of various dependent variables (e.g., unitary costs, unitary task times, quality metrics) that are affected by experience and can be described in terms of autonomous ('learning by doing') or induced (e.g., training hours, investments, equipment) learning sources. Quality metrics (e.g., Lolli et al. 2016a; Ittner et al., 2001; Lolli et al. 2018), task times (e.g., Biskup 1999; Bailey 1989), and costs (e.g., Lolli et al. 2016b) are the most prevalent dependent variables utilized in learning curves in a variety of industrial and service operations.

Assembly lines, which require a lot of manual labor, are an example of a manufacturing environment where learning effects are essential in determining task time. The goal of balancing an assembly line is to assign assembly tasks to workstations in order to maximize a specific performance measure while adhering to precedence constraints. The large variety of assembly line balancing problems (ALBP) are based on two basic models and named as type I and type II models. The type I ALBP (e.g., Li et al. 2017b; Gansterer and Hartl 2017) aims at minimizing the number of workstations required to meet a specific cycle time, whereas the type II ALBP (e.g., Tang et al. 2016; Li et al. 2017a) aims at minimizing the cycle time for a fixed number of workstations (Baybars, 1986). Furthermore, Boysen et al. (2007) divided ALBPs into three categories: single-model ALPBs, mixed-model ALPBs (where one product is created in different models on the same assembly line), and multi-model ALPBs (where multiple products are manufactured in batches). The nature of task times, either deterministic or stochastic, is another factor used to classify ALBPs. For a review of this topic, see Battaia and Dolgui (2013). ALBPs that use stochastic methods work effectively in labor-intensive assembly lines where task times are assumed to be normally distributed. However, as a result of experience, the expected values of these task times decrease. That is to say, the lower the task times, the higher the quantity of assembled products. Because of the decreasing task times as a result of experience, the best balancing methods evolve with time. Our proposal fits under the deterministic type I ALBP with the learning effects research stream.

## 1.1 Objectives

The main novelty of this paper is that when a simple assembly line balancing problem type1(SALP-1) is solved the change of task times as a consequence of learning is considered. Two LC models were chosen: The Wright's power function and one of its extensions, the plateau LC model. In each case, the minimum number of workstations (WSs) and the corresponding optimal allocation of tasks and operators to workstations are determined. Consequently, the main contributions of the paper are twofold:

- A modified model is elaborated, which can be used to determine the minimum number of workstations when the learning effect is taking place.
- The effect of the change of the LC models and the learning rates on the optimal solutions for a sample problem are also illustrated and examined.

In this paper, the effect of two widely used learning curves, the Wright's learning function and the plateau model (see, for instance, Vits & Gelders, 2002 or Anzanello & Fogliatto, 2011), are applied. However, the proposed ALB model can be extended for any other learning curves.

The remainder of this paper is structured as follows. Section 2 presents a literature review of most relevant papers related to the presented research. Section 3 briefly presents the LC models applied. Section 4 outlines the formulation of the modified (SALBP-1) model. Section 5 illustrates the application of presented method on a sample problem and discusses the obtained results. Finally, the main results are summarized, and some future research possibilities are outlined.

## 2. Literature Review

Cohen and Dar-El (1998) offered the first contribution on learning effects in assembly lines, in which a type I ALBP with deterministic task times was solved analytically. Cohen et al. (2006) used the Wright's learning curve (1936) with homogeneous learning slopes between workstations and predictable task times to examine the inverse II ALBP. Toksari et al. (2008) used the position-dependent learning curve described by Biskup to deal with a type I ALBP (1999). They showed that simple and U-type line balancing problems with homogeneous learning effect can be solved with polynomial algorithms. Toksari et al. (2010) used a mixed nonlinear integer programming model to study type I ALBP. The Biskup's learning curve (1999) was combined with a linear increase in task time owing to work deterioration in this case. Hamta et al. (2013) developed a meta-heuristic approach for a deterministic multi-objective ALBP, based on Biskup's learning curve (1999) to describe position-dependent task times.

In the context of learning, Koltai et al. (2015) proposed an algorithm for determining the throughput time of a simple assembly line. They showed that, while classical ALB models assume a constant cycle time, cycle time in the case of learning might change for two reasons. First, according to the station time function, cycle time decreases exponentially with learning. Second, the bottleneck could move from station to station, causing the cycle time to alter even further.

Lolli et al. (2017) examined a stochastic type I ALBP with learning effects, using the cost-based Kottas-Lau heuristic (1973) and the well-known Wright's curve (1936) with a plateau. Learning affects the optimal balancing solution over time; therefore, the assembly line must be rebalanced. The rebalancing problem was applied as a consequence of the learning process involving each assembly workstation in Lolli et al. (2017). Rebalancing, however, might be necessary due to the changes in market conditions or product design as well (Gamberini et al. 2006; Gamberini et al. 2009).

## 3. Learning curve models

In the problem presented in this paper, we assumed that the task time decreases due to the learning effect, which means that the more frequently a task is performed by a human worker, the less time it takes to be performed. Since the Wright and the plateau LC models are the most used in practice (Vits & Gelders 2002 or Anzanello & Fogliatto, 2011), we suppose that the task time decreases according to these two models.

### 3.1 Wright's model

The Wright model is generally viewed as the basic learning curve (LC) in the literature (Figure1). Wright (1936) studied aircraft assembly processes and noticed that as the number of units produced doubles, the time required to manufacture each unit decreases at a steady rate. The mathematical formulation of the Wright LC is as follows:

$$Y(Q) = a \cdot Q^b$$

where  $Y$  is the average time (or cost) per unit required to produce  $Q$  units,  $a$  is the time (cost) to produce the first unit, and  $Q$  is the cumulative number of units. Parameter  $b$  ( $-1 < b < 0$ ) is the slope of the learning curve, which describes the workers' learning rate. Values of  $b$  close to  $-1$  indicate a high learning rate and fast adaptation to task execution (Teplitz 1991, Badiru, 1992, Argote, 1999, Dar-El 2000).

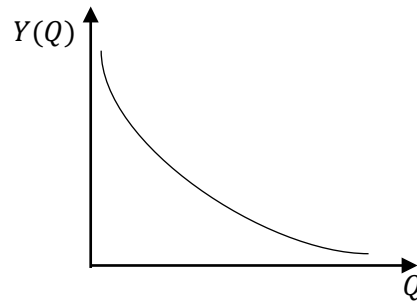


Figure 1. Wright's learning curve model

### 3.2 Plateau model

The plateau model completes Wright's model by introducing a constant  $C$  to overcome the problem of zero time/cost at large production quantities (Figure 2). The constant  $C$  denotes a phenomenon known as plateauing, which indicates that the learning effect is finite (Baloff, 1971). This model suggests that the unit time/cost can decrease until reaching the steady-state level  $Q_s$ , after which the unit time/cost becomes constant. The mathematical formulation of the Plateau LC is as follows:

$$Y(Q) = \begin{cases} a \cdot Q^b & \text{if } Q < Q_s \\ C & \text{if } Q \geq Q_s \end{cases}$$

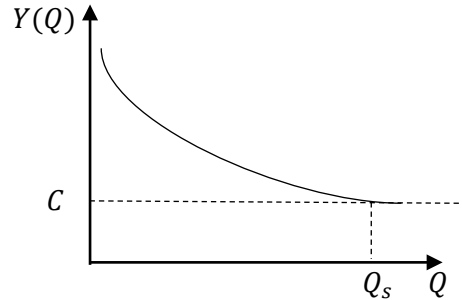


Figure 2. Plateau learning curve model

### 4. Formulation of the modified SALB-1 model

In this section, a modified SALBP-1 model which incorporates the effect of learning is presented. The applied notations are summarized in table 1. Tasks are numbered in a continuously increasing order. The number  $i$  assigned to a task is called the task index. We refer to a task either by its name or task index. Those tasks which are not succeeded by any other task are called last tasks. The index set of last tasks is denoted by  $F$ .

Workstations are also numbered in a continuously increasing order. The first workstation is numbered 1 and the last workstation is numbered  $J$ . The number  $j$  assigned to a workstation is called the workstation index. Workstations are referred to in this paper by the workstation index. Before task assignment, an assumption must be made about the possible number of stations. Similarly, workers are numbered in a continuously increasing order. We refer to a worker by its index  $k$ , which is a number between 1 and  $K$ .  $K$  is the number of available workers.

The assignment of tasks and workers to workstations is expressed with the  $x_{ijk}$  binary decision variable. If task  $i$  and worker  $k$  are assigned to the same workstation  $j$ , then  $x_{ijk} = 1$ , otherwise  $x_{ijk} = 0$ . Similarly, the assignment of workers

to workstations is expressed with the  $y_{jk}$  binary decision variable. If worker  $k$  is assigned to workstation  $j$ , then  $y_{jk} = 1$ , otherwise  $y_{jk} = 0$ . The following integer linear programming formulation of the modified SALBP-1 is used in this paper,

$$\begin{aligned} & \text{Min } (N) && (1) \\ & \sum_{i=1}^I \sum_{k=1}^K t_{ik} x_{ijk} \leq T_c && \forall j \quad (2) \\ & \sum_{j=1}^J \sum_{k=1}^K x_{ijk} = 1 && \forall i \quad (3) \\ & \sum_{j=1}^J j(x_{qjk} - x_{pjk}) \geq 0 && \forall (p, q) \in R \quad (4) \\ & x_{ijk} \leq y_{jk} && \forall i, j, k \quad (5) \\ & \sum_{j=1}^J y_{jk} \leq 1 && \forall k \quad (6) \\ & \sum_{k=1}^K y_{jk} \leq 1 && \forall j \quad (7) \\ & \sum_{j=1}^J \sum_{k=1}^K j x_{ijk} \leq N && \forall i \in F \quad (8) \end{aligned}$$

The objective of the model is to constantly minimize the number of workstations  $N$  while the workers' learning effect is taking place. As a worker  $k$  performs a certain task  $i$ , the corresponding task time  $t_{ik}$  starts decreasing according to the learning curve depending on how many times the task has been performed. Therefore, the task times  $t_{ik}$  are constantly updated for each worker after each unit produced. Task times keep decreasing until fewer workstations are required, and consequently, tasks and operators are differently allocated.

Minimizing the number of workstations can be done by minimizing the largest index ( $N$ ) pertaining to a station with task assignment. The left-hand side of constraint (8) determines the index of those workstations which perform the last tasks. The highest such index must be minimized. If each of these indices is smaller than or equal to  $N$ , and  $N$  is minimized, then the index of the final workstation, and hence the number of workstations, is minimized.

Cycle time constraints are expressed by constraint (2). For each workstation  $j$  at which a worker  $k$  performs a certain task(s)  $i$ , the sum of task times of the assigned tasks is not allowed to exceed the predefined cycle time  $T_c$ . Constraint (3) ensures that each task is performed by one of the workers at a workstation. Precedence constraints are expressed by constraint (4). Since task  $p$  must be assigned to an earlier or to the same workstation as task  $q$ ; the weighted sum of these differences is always higher than or equal to 0 if the weights are the indices of the corresponding workstations.

The assignment constraints are expressed by constraints (5), (6), and (7). Constraint (5) ensures that to each workstation  $j$  where a task  $i$  is performed, a worker  $k$  must be assigned. Constraint (6) makes sure that each worker  $k$  can be assigned at most to one of the workstations. Finally, Constraint (7) ensures that to a workstation  $j$ , at most one worker is assigned.

Table 1. Summary of notations applied in the model

---

**Summary of notations:**

---

*Indices:*

$i$  = index of tasks ( $i=1, \dots, I$ ),  
 $p$  = index of subtasks,  
 $q$  = index of subtasks,  
 $j$  = index of workstations ( $j=1, \dots, J$ ),  
 $k$  = index of workers ( $k=1, \dots, K$ ),

*Parameters:*

$I$  = number of tasks,  
 $J$  = maximum number of workstations,  
 $N^*$  = minimum number of stations (the result of the station number minimization of the basic SALBP-1 model),  
 $K$  = number of available workers ( $K > N^*$ ),  
 $t_i$  = initial task time of task  $i$ ,  
 $Q_{ik}$  = rank of the part on which task  $i$  is performed by worker  $k$ ,  
 $Q_s$  = rank of the part at which the learning effect stops for the workers (Plateau LC model),  
 $t_{ik}$  = time necessary to perform task  $i$  by worker  $k$  on  $Q_{ik}$ ,  
 $b$  = power of the learning curve function,  
 $L$  = learning rate ( $L = 2^b$ ),

*Sets:*

$F$  = set of final tasks,  $i \in F$ , if task  $i$  does not precede any other task,  
 $R$  = set of pair of indices which belong to tasks with precedence relations, that is,  $(p; q) \in R$ , if task  $p$  immediately precedes task  $q$ ,

*Decision variables:*

$N$  = objective function variable for the number of workstations,  
 $x_{ijk}$  = 0-1 decision variable, if  $x_{ijk}=1$ , then task  $i$  is performed by worker  $k$  at workstation  $j$ , otherwise  $x_{ijk}=0$ ,  
 $y_{jk}$  = 0-1 decision variable; if  $y_{jk}=1$ , then worker  $k$  is assigned to workstation  $j$ , otherwise  $y_{jk}=0$ ,

---

**Case 1: Wright’s LC model**

$$t_{ik} = t_i \cdot (Q_{ik})^b$$


---

**Case 2: Plateau LC model**

$$t_{ik} = \begin{cases} t_i \cdot (Q_{ik})^b & \text{if } Q_{ik} < Q_s \\ t_i \cdot (Q_s)^b & \text{if } Q_{ik} \geq Q_s \end{cases}$$

**5. Practical illustration of the presented model**

**5.1 Problem description**

To illustrate the performance of the presented model, let us consider a simple example taken from the textbook “Production and operations management: Manufacturing and services” by Richard B. Chase and Nikolas J. Aquilano.

The table shows the list of tasks of the assembly, the immediately preceding tasks, and the task times. Based on the information of Table 2, the precedence graph of tasks can be easily depicted (see Figure 3).

Table 2. Assembly tasks and task times for Model J Wagon

<i>i</i>	Time of task <i>i</i> (s)	Tasks	Immediate precedent of task <i>i</i>
A	45	Position rear axle support and hand fasten four screws to nuts.	-
B	11	Inset rear axle.	A
C	9	Tighten rear axle support screws to nuts.	B
D	50	Position front axle assembly and hand fasten with four screws to nuts.	-
E	15	Tighten front axle assembly screws.	D
F	12	Position rear wheel #1 and fasten hubcap.	C
G	12	Position rear wheel #2 and fasten hubcap.	C
H	12	Position front wheel #1 and fasten hubcap.	E
I	12	Position front wheel #2 and fasten hubcap.	E
J	8	Position wagon handle shaft on front axle assembly and hand fasten bolt and nut.	F, G, H, I
K	9	Tighten bolt and nut.	J

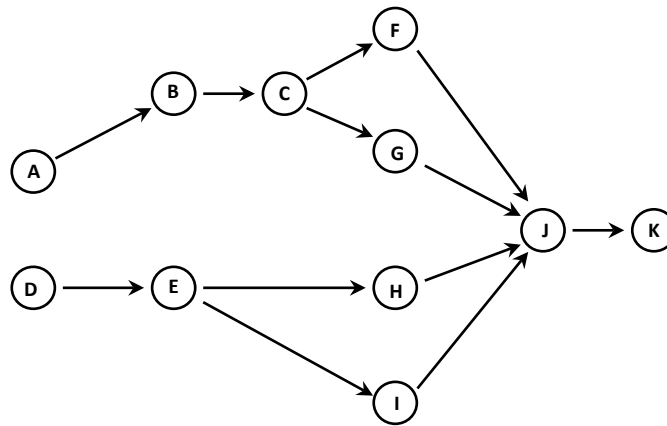


Figure 3. Precedence relations between tasks

500 Wagons are required daily, and the production time is 420 minutes per day. Based on these data, the required cycle time is 50.4 seconds ( $420 \times 60/500$ ). The objective is to minimize the number of workstations while taking the learning effect into consideration without the deterioration of the predefined cycle time.

### 5.2 Modified SALBP-1 model application

The effect of learning on the optimal number of stations is analyzed with two learning curve models: Wright’s LC model and Plateau LC model. For each of the LC models, four learning rates are chosen: 0.6, 0.7, 0.8, and 0.9, which gives the corresponding four values of *b* respectively: -0.74, -0.51, -0.32, and -0.15. We note that any number of operators *K* equal to or higher than the minimum number of WSs can be chosen. Since in this example the minimum number of WSs is 4, we opted for a *K* value of 4.

For the sake of simplicity, we assume that work-in-process inventory cannot accumulate between stations. This implies that the worker assigned to station (*j* + 1) cannot start his task(s) unless the worker assigned to station *j* has finished the task(s).

The solution of the modified SALBP-1 model defined for each of the two learning curves requires a flexible mathematical modeling tool. Such a tool is provided by the AIMMS Prescriptive Analytics Platform, which is often applied for solving commercial optimization problems (Roelofs and Bisschop 2018). AIMMS offers a straightforward

mathematical modeling environment and a wide range of available solvers, including CPLEX to solve LP problems. In this research, AIMMS version 4.84 was used to create the required mathematical models, implement the algorithms, and create a simple user interface. CPLEX version 12.7.1 was used to solve the generated LP.

To monitor the evolution of the minimum number of workstations as the learning effect occurs for the workers, we solved the model after the production of each unit, which means that when a unit exits the last station the station times updated according to the learning curve and the problem is resolved. Since the demand in the example is 500 units per day, we repeated the solution of the models 500 times.

Figure 4 shows the minimum number of WSs in function of the output units  $Q$  in the case of Wright’s LC. We notice that for all the four learning rates ( $L=0.6, 0.7, 0.8, 0.9$ ), the minimum number of workstations after the production of the first unit ( $Q = 1$ ) is 4, which is the same minimum number of stations that can be obtained if the basic SALPB-1 model is solved. At this stage, the workers have not progressed yet along the learning curve, consequently, the learning effect has not started yet.

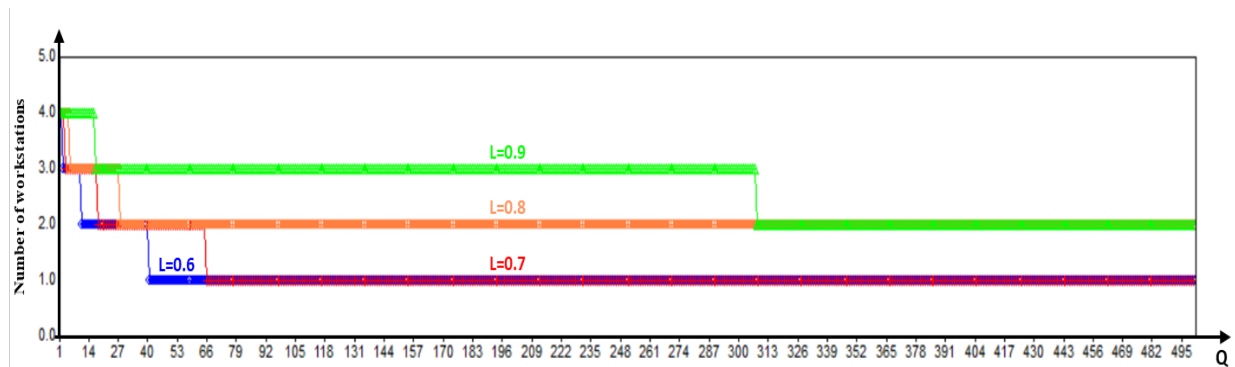


Figure 4. Tracking of the minimum number of WSs in function of the output units in case of the Wright’s model

As the workers repeat the assigned tasks, the task times decrease exponentially. The decrease of the task times for the allocated workers causes the decrease of the station times as well. The drop of the station times continues until fewer stations are required to satisfy the predetermined cycle time constraint.

The minimum number of WSs keeps decreasing until we reach one workstation, which means that all the tasks are assigned to only a single station. For instance, for  $L=0.6$ , the number of workstations decreases from 4 to 3 at  $Q=2$ , then to 2 at  $Q=10$ , and finally to 1 at  $Q=40$ . Similarly, for  $L=0.7$ , the number of workstations drops from 4 to 3 at  $Q=3$ , then to 2 at  $Q=17$ , and finally to 1 at  $Q=65$ . We know that the higher the learning rate is, the slower the decrease of the station times, that is why for  $L=0.8$  and  $0.9$ , we could not reach the single workstation optimum for a maximum output of 500 units. A single station optimum could be reached if we opted for a higher number of output units.

Table 3 summarizes the allocation of the tasks and operators to the minimum number of WSs and the corresponding station times at the output units where the minimal number of WSs changes in the case of the Wright’s LC model for the four chosen learning rates. We note that at the start of production, when  $Q=1$ , for all the four learning rates, the minimal number of stations, tasks, operators, and station times is the same since the learning effect has not begun yet. However, after the first unit, the task times start constantly decreasing until fewer WSs are required. Consequently, the tasks and operators are differently allocated, as the table shows.

Table 3. Minimum number of WSS, the allocated tasks and operators, and the station times in case of Wright’s LC model

	Learning rate(L)															
	0.6				0.7				0.8				0.9			
Output unit(Q)	1				1				1				1			
Stations	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
tasks	D	A	B,E,H,I	C,F,G,J,K	D	A	B,E,H,I	C,F,G,J,K	D	A	B,E,H,I	C,F,G,J,K	D	A	B,E,H,I	C,F,G,J,K
Operators	O1	O2	O3	O4	O1	O2	O3	O4	O1	O2	O3	O4	O1	O2	O3	O4
Station time(s)	50.00	45.00	50.00	50.00	50.00	45.00	50.00	50.00	50.00	45.00	50.00	50.00	50.00	45.00	50.00	50.00
Output unit(Q)	2				3				5				16			
Stations	1	2	3		1	2	3		1	2	3		1	2	3	
tasks	A,B,C	D,E	F,G,H,I,J,K		A,B,C	D,E	F,G,H,I,J,K		D,E	A,B,H	C,F,G,I,J,K		A,B,C	D,F	E,G,H,I,J,K	
Operators	O2	O1	O4		O2	O1	O4		O1	O2	O4		O2	O1	O4	
Station time(s)	47.0	45.0	48.6		45.57	43.4	47.3		44.8	49.8	43.5		49.5	49.2	50.4	
Output unit(Q)	10				17				27				307			
Stations	1	2			1	2			1	2			1	2		
tasks	A,B,C,F,G	D,E,H,I,J,K			A,B,C,F,G	D,E,H,I,J,K			D,E,H,I	A,B,C,F,G,I,K			A,B,C,F,G	D,E,H,I,J,K		
Operators	O2	O1			O2	O1			O1	O2			O2	O1		
Station time(s)	40.8	49.4			33.9	47.5			36.3	49.2			41.6	50.4		
Output unit(Q)	40				65											
Stations	1				1											
tasks	A,B,C,D,E,F,G				A,B,C,D,E	F,G,H,I,J,K										
Operators	O1				O4											
Station time(s)	43.4				49.8											

Figure 5 shows the minimum number of workstations in function of the output units  $Q$  in the case of the Plateau LC model. The same remarks mentioned before about the Wright’s LC case can be valid for this case as well, with the exception that we set a  $Q_s$  of 100 units. This means that when a worker repeats a certain task 100 times, the learning effect for this worker about this task stops, and the plateau is reached.

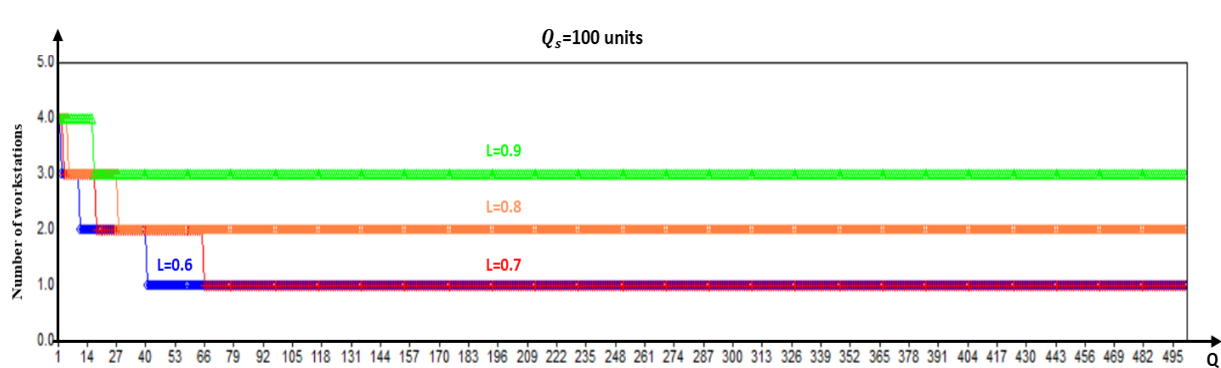


Figure 5. Tracking of the minimum number of WSS in function of the output units in case of the Plateau model LC

For the three learning rates 0.6, 0.7, and 0.8, the minimal number of WSS drops precisely at the same output points as for the Wright’s LC model, because all the drops occurred before  $Q_s$ , and the Plateau does not affect the results since none of the workers could repeat a task more than 100 times. However, for  $L=0.9$ , we could not reach two workstations as in the previous case due to slow learning. In the Wright’s LC case, the shift from two to three workstations occurred at  $Q=307$ . In this case the plateauing effect influences the final result.

Table 4 summarizes the allocation of the tasks and operators to the minimum number of WSS and the corresponding station times at the output units where the minimal number of WSS changes in the case of the Plateau LC model for the four chosen learning rates. The results are identical to the Wright’s LC model for the four learning rates, because the minimal number of WSS’ drops happened before  $Q_s$ . The Plateauing effect could not influence the allocated tasks and operators and consequently the station times in these cases. If, however,  $Q_s$  is set below the output quantity at which the minimal number of WSS decreases, the results might be completely different.



Table 4. Minimum number of WSs, the allocated tasks and operators, and the station times in the case of Plateau LC model

	Learning rate(L)															
	0.6				0.7				0.8				0.9			
Output unit(Q)	1				1				1				1			
Stations	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
tasks	D	A	B,E,H,I	C,F,G,J,K	D	A	B,E,H,I	C,F,G,J,K	D	A	B,E,H,I	C,F,G,J,K	D	A	B,E,H,I	C,F,G,J,K
Operators	O1	O2	O3	O4	O1	O2	O3	O4	O1	O2	O3	O4	O1	O2	O3	O4
Station time(s)	50.00	45.00	50.00	50.00	50.00	45.00	50.00	50.00	50.00	45.00	50.00	50.00	50.00	45.00	50.00	50.00
Output unit(Q)	2				3				5				16			
Stations	1	2	3		1	2	3		1	2	3		1	2	3	
tasks	A,B,C	D,E	F,G,H,I,J,K		A,B,C	D,E	F,G,H,I,J,K		D,E	A,B,H	C,F,G,I,J,K		A,B,C	D,F	E,G,H,I,J,K	
Operators	O2	O1	O4		O2	O1	O4		O1	O2	O4		O2	O1	O4	
Station time(s)	47.0	45.0	48.6		45.57	43.4	47.3		44.8	49.8	43.5		49.5	49.2	50.4	
Output unit(Q)	10				17				27							
Stations	1	2			1	2			1	2						
tasks	A,B,C,F,G	D,E,H,I,J,K			A,B,C,F,G	D,E,H,I,J,K			D,E,H,I	A,B,C,F,G,I,K						
Operators	O2	O1			O2	O1			O1	O2						
Station time(s)	40.8	49.4			33.9	47.5			36.3	49.2						
Output unit(Q)	40				65											
Stations	1				1											
tasks	A,B,C,D,E	F,G,H,I,J,K			A,B,C,D,E	F,G,H,I,J,K										
Operators	O1				O4											
Station time(s)	43.4				49.8											

We can conclude that there are two distinct cases:

- If the minimum number of WSs decreases before  $Q_s$  is reached in the Wright’s LC model, the minimum number of WSs, the allocated tasks and operators, and the station times will be identical to the results of the Plateau LC model.
- However, If the minimum number of WSs decreases after  $Q_s$  in case of the Wright’s LC model, the allocated tasks, and operators and consequently the station times would be different in case of the Plateau LC model, and the decrease of the minimal number of WSs could be delayed or never be reached.

## 6. Conclusion

In simple assembly lines without the learning effect, the task times are constant, and consequently, so are the station times; therefore, when solving the SALBP-1 model, merely one solution is found. In the presence of the learning effect, as tasks are repeated frequently, task times and station times decrease, causing the minimal number of workstations to drop along with the output units and the SALBP-1 has more than one solution.

In this paper, a modified SALBP-1 model incorporating the learning effect is formulated and applied. The modified SALBP-1 model developed in section 4 is illustrated with a simple example in section 5. The Wright’s and the Plateau learning curves were chosen for our model because of their wide use. The model, however, can be valid for any other learning curves.

In the illustrative example, two learning curves were chosen with four different learning rates: 0.6, 0.7, 0.8, and 0.9. The results showed that the minimal number of workstations at the start of production ( $Q=1$  units) is the same as the basic SALBP-1 model (In our case, it was 4), which is logical since the learning effect has not started yet. However, for both LC models, the minimal number of WSs decreases along with the output unit due to the learning effect causing the allocation of tasks and operators and consequently the station times to change after each produced unit.

Comparing the two learning curves, the results showed that if the decrease of the minimal number of workstations in the case of Wright’s LC model occurs before the plateauing quantity  $Q_s$ , the results will be identical in the case of the Plateau LC model regardless of the learning rate. Nonetheless, if it happens after  $Q_s$ , then the decrease of WSs in case of the Plateau LC model can be either delayed or never reached.

In the present paper, the evolution of the minimal number of workstations in the presence of the learning effect is investigated. As a result, different task allocations and station number configurations were obtained during a production run. In practice, it is not practical and frequently not even feasible to change line configuration frequently during a production run. The solution for this problem is a unique line configuration that considers the evolution of task time in the early stages of operation when the decrease of task time is significant. How to obtain the optimal task allocation, which considers the change of task times as a consequence of learning, and also the stability requirement

of task allocation during the whole production run is a challenging topic for further research. The proposed challenging research problem, however, cannot be addressed without the knowledge of the change of optimal assignment explained by the change of task times. This information is generated by the calculation proposed in this paper.

### References:

- B. Chase, R., and J. Aquilano, N., *Production and Operations Management: Manufacturing and Services*, 7<sup>th</sup> edition, Richard D Irwin, 1995.
- Battaia, O., and Dolgui, A., A taxonomy of line balancing problems and their solution approaches, *International Journal of Production Economics*, vol. 142, no. 2, pp. 259-277, 2013.
- Baybars, I., A survey of exact algorithms for the simple assembly line balancing problem, *Management science*, vol. 32, no. 8, pp. 909-932, 1986.
- Bentaha, M. L., Dolgui, A., and Battaia, O., A bibliographic review of production line design and balancing under uncertainty, *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 70-75, 2015.
- Biskup, D., Single-machine scheduling with learning considerations, *European Journal of Operational Research*, vol. 115, no. 1, pp. 173-178, 1999.
- Boysen, N., Fliedner, M., and Scholl, A., A classification of assembly line balancing problems, *European journal of operational research*, vol. 183, no. 2, pp. 674-693, 2007.
- Chica, M., Bautista, J., Cerdón, Ó., and Damas, S., A multiobjective model and evolutionary algorithms for robust time and space assembly line balancing under uncertain demand, *Omega*, vol. 58, pp. 55-68, 2016.
- Chutima, P., and Naruemitwong, W., A Pareto biogeography-based optimisation for multi-objective two-sided assembly line sequencing problems with a learning effect, *Computers & Industrial Engineering*, vol. 69, pp. 89-104, 2014.
- Cohen, Y., and Ezey Dar-El, M., Optimizing the number of stations in assembly lines under learning for limited production, *Production Planning & Control*, vol. 9, no. 3, pp. 230-240, 1998.
- Cohen, Y., Vitrer, G., and Sarin, S. C., Optimal allocation of work in assembly lines for lots with homogenous learning, *European Journal of Operational Research*, vol. 168, no. 3, pp. 922-931, 2006.
- Dolgui, A., Gordon, V., and Strusevich, V., Single machine scheduling with precedence constraints and positionally dependent processing times, *Computers & Operations Research*, vol.39, no. 6, pp. 1218-1224, 2012.
- Gamberini, R., Gebennini, E., Grassi, A., and Regattieri, A., A multiple single-pass heuristic algorithm solving the stochastic assembly line rebalancing problem, *International Journal of Production Research*, vol. 47, no. 8, pp. 2141-2164, 2009.
- Hamta, N., Ghomi, S. F., Jolai, F., and Bahalke, U., Bi-criteria assembly line balancing by considering flexible operation times, *Applied Mathematical Modelling*, vol. 35, no. 12, pp. 5592-5608, 2011.
- Jaber, M. Y., and Glock, C. H., A learning curve for tasks with cognitive and motor elements, *Computers & Industrial Engineering*, vol. 64, no. 3, pp. 866-871, 2013.
- Koltai, T., and Kalló, N., Analysis of the effect of learning on the throughput-time in simple assembly lines, *Computers & industrial engineering*, vol. 111, pp. 507-515, 2017.
- Li, Z., Tang, Q., and Zhang, L., Two-sided assembly line balancing problem of type I: Improvements, a simple algorithm and a comprehensive study, *Computers & Operations Research*, vol. 79, pp. 78-93, 2017.
- Lolli, F., Messori, M., Gamberini, R., Rimini, B., and Balugani, E., Modelling production cost with the effects of learning and forgetting, *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 503-508, 2016.
- Salveson, M. E., The assembly line balancing problem, *The journal of industrial engineering*, vol.6, no. 3, pp. 18-25, 1955.
- Toksarı, M. D., İşleyen, S. K., Güner, E., and Baykoç, Ö. F., Assembly line balancing problem with deterioration tasks and learning effect, *Expert systems with Applications*, vol. 37, no. 2, pp. 1223-1228, 2010.
- Toksarı, M. D., İşleyen, S. K., Güner, E., and Baykoç, Ö. F., Simple and U-type assembly line balancing problems with a learning effect, *Applied Mathematical Modelling*, vol. 32, no. 12, pp. 2954-2961, 2008.
- Wright, T. P., Factors affecting the cost of airplanes, *Journal of the aeronautical sciences*, vol. 3, no. 4, pp. 122-128, 1936.

### Acknowledgement:

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

### **Biographies:**

**Zakaria Zine El Abidine** is currently a Ph.D. student at Budapest University of Technology and Economics. He has two master's degrees, the first in industrial engineering from the faculty of sciences and techniques (FST) of Fes in Morocco in 2018 and the second in engineering management from the University of Debrecen in Hungary in 2020. His research interests include assembly line balancing, learning curves, process optimization, and process mining.

**Imre Dimény** is a PhD candidate at the Budapest University of Technology and Economics with a master's degree in applied mathematics from the Eötvös Loránd University and an MBA degree from the Budapest University of Technology and Economics. His research interests include assembly line balancing, linear programming, and mathematical modelling.

**Tamás Koltai** is a professor of production and operations management and currently Dean of the Faculty of Economic and Social Sciences at the Budapest University of Technology and Economics. He has a master's degree in mechanical engineering and a PhD in industrial engineering. He is head of the Production Management Group of his department. He also works as editor in chief of Periodica Polytechnica Social and Management Sciences series. His research area includes mathematical modeling of manufacturing and service systems and performance evaluation of production and service operations. Orcid: <http://orcid.org/0000-0001-6873-6944>.