



HAL
open science

Comparing linear and non-linear modeling approaches of learning effects in 2-stage flowshop scheduling problems

Yenny A. Paredes-Astudillo, Valérie Botta-Genoulaz, Jairo R. Montoya-Torres

► To cite this version:

Yenny A. Paredes-Astudillo, Valérie Botta-Genoulaz, Jairo R. Montoya-Torres. Comparing linear and non-linear modeling approaches of learning effects in 2-stage flowshop scheduling problems. 10th IFAC Conference on Manufacturing Modelling, Management and Control (MIM), Jun 2022, Nantes, France. hal-03824322

HAL Id: hal-03824322

<https://hal.science/hal-03824322>

Submitted on 20 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Comparing linear and non-linear modelling approaches of learning effects in 2-stage flow-shop scheduling problems

Yenny Alexandra Paredes-Astudillo^{*,**}, Valérie Botta-Genoulaz^{*}, Jairo R. Montoya-Torres^{**}

^{*} Univ Lyon, INSA Lyon, Université Claude Bernard Lyon 1, Univ Lyon 2, DISP-UR4570, 69621 Villeurbanne, France (e-mail : yenny.paredes-astudillo@insa-lyon.fr , valerie.botta@insa-lyon.fr)

^{**} School of Engineering, Universidad de La Sabana, km 7 autopista norte de Bogotá D.C., Chía, Colombia (e-mail: yennypaas@unisabana.edu.co , jairo.montoya@unisabana.edu.co)

Abstract: Scheduling problems deal with the allocation and sequencing of a set of jobs to a set of production resources (machines, workers) to optimize one or more objective functions. In the literature, traditional deterministic approaches consider that processing times of jobs are fixed, constant and known at the initial time of scheduling. However, this assumption is not realistic particularly in hand-intensive manufacturing, where processing times may vary according to workers' learning about the production process. To deal with this, this paper considers the problem of jobs scheduling with learning effect in the processing times. In contrast with previous works in the literature, it is not the processing time that changes *per se*, but the resource (worker) that improves his/her productivity by decreasing the actual job processing time. This paper considers the flow-shop configuration in which all jobs are processed by a set of different workers following the same production routing. It reviews linear and non-linear modelling approaches of the learning effect in processing times, in order to gain insights about the actual impact of such different modelling approaches. A computational comparison of exact solutions based on mathematical modelling of the flow-shop problem is presented for the case of two workers. Experiments were run using random-generated instances. Several insights are drawn regarding the behavior of the different models.

Copyright © 2022 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: flow-shop, scheduling, learning effect, mathematical modelling, performance analysis.

1 INTRODUCTION

Despite the current industrial development and technological advances worldwide (Boudreau et al., 2003), people are key to decision-making and the operation of any system (Neumann and Dul, 2010). Moreover, although automation has increased in many production systems, there are tasks that do require the intervention of humans (Sánchez-Herrera et al., 2019). Indeed, as pointed out by Heizer and Render (2013), operations experts recognize the direct or indirect importance of people as key resources in productive systems. However, it is clear that there is a separation between human factors and operation management; Neumann and Dul (2010) indicate in their review that human well-being publications are rarely published in business or management journals, even though the execution of a task may have effect on the system and humans (Lodree et al., 2009, Sánchez-Herrera et al., 2019; Bentefouet and Nembhard, 2013).

One human process that directly impacts on productivity is the learning effect. Thus, the benefits of including accurate labor standards allows to measure realistic production objective (Glock et al. 2019). In order to study the impact of worker behavior on the schedule, this paper consider the flow-shop scheduling problem with learning effect. It is built upon previous works (e.g., Moreno-Camacho et al., 2018, Ruiz-Torres et al., 2013) in which the processing time of jobs do not change *per se*, but instead, the resources (workers) are

impacted by the learning effect. So, while it is true that this is an issue that has been addressed in recent literature, measuring the performance of manual activities within companies is a challenge that comes from the appearance of productive systems. But identifying an accurate approach for phenomena such as learning would provide precise standards for comping the processing times in modelling labor-intensive manufacturing systems, such as textiles, picking process, or luxury industries (Pinedo, 2018; Sánchez-Herrera et al., 2019; Montoya-Torres et al., 2021).

In recent years, the need of understanding and model different human phenomena and characteristics has grown. In the case of the learning effect, most approaches emerged following the work of Wright (1936), which presents one of the most common equations for calculating the learning effect in manufacturing systems. However, the first studies on scheduling with learning only appear about 20 years ago, with Biskup (1999) and Cheng & Wang (2000) whose articles became the baseline on the area. These studies were focusing on the single-machine configuration, the first one dealt with the position-based learning effect, while the latter considered volume-based learning using a linear piecewise function for the processing time.

In general terms, the most common approaches are the learning effect based on either the position in the schedule or the sum-of-processing-times. Some authors have made

variations including a control parameter resulting in the truncated learning effect for both cases.

The approach used to model the learning effect in a flow-shop scheduling problem may affect the proximity to the problem reality and computational complexity. Thus, it is the case of models based on the sum-of-processing-time, which are considered closer to human behavior (Kuo & Yang, 2006), but correspond to non-linear models with higher computational complexity.

Most articles on this subject develop methods to solve the problem or analyze the complexity of the problem. However, no work was found comparing different learning effect approaches and their effect on the objective function and CPU time.

The objective of this paper is twofold. On one hand, it reviews the different approaches employed so far in the literature to model the learning effect in flow-shop scheduling; on another hand, it seeks to experimentally evaluate the impact of such learning effect modelling approaches. By doing so, we will gain insights about the actual impact and benefits of using one approach or another. This will also aid decision-making processes in practice when defining operations scheduling in hand-intensive manufacturing systems. This paper theoretically addresses the problem with benchmark data sets; in the future, these models could be also applied to real-life contexts. For instance, by including case studies, production can be planned according to workers' training strategies.

The rest of the paper is organized as follows. The different approaches to model the learning effect in flow-shops are given in Section 2, while the experimental setting and analyses of the results are presented in Section 3. Finally, Section 4 summarizes the conclusions of this work and outlines some opportunities for future research.

2 PROBLEM STATEMENT AND MATHEMATICAL MODELS FOR THE LEARNING EFFECT

2.1 The flow-shop scheduling problem

Formally speaking, this paper considers the problem of scheduling a set of n independent jobs, $j = 1, \dots, n$, to be processed by a set of m workers, $i = 1, \dots, m$. All jobs must follow the same production routing, workers are organized in series: job j is executed first by worker 1, then by worker 2, ..., and finally by worker m . Each worker can process one job at a given time and preemption of a job is not allowed (i.e., the execution of a job cannot be interrupted once its processing has started). All workers are available at the initial time of scheduling (production horizon) and have a 100% production rate. The processing time of job j does depend on the worker i who executes it (\bar{P}_{ij}). The flow-shop scheduling problem can be modeled as a mixed-integer linear programming (MILP) model. The definition and notations are shown below:

Sets

I : workers
 J : jobs
 R : positions

Parameters

\bar{P}_{ij} : processing time of job j executed by worker i
 M : represents a very big number or Big M

Decision Variables:

X_{jr} : $\begin{cases} 1: & \text{if the job } j \text{ is processed in position } r \\ 0: & \text{otherwise} \end{cases}$
 C_{ij} : completion time of job j on worker i
 C_{max} : makespan value

Objective function

$$\text{Minimize } Z = C_{max} \tag{1}$$

Subject to:

$$\sum_{r \in R} X_{jr} = 1 \quad \forall j \in J \tag{2}$$

$$\sum_{j \in J} X_{jr} = 1 \quad \forall r \in R \tag{3}$$

$$C_{1j} \geq \bar{P}_{1j} \quad \forall j \in J \tag{4}$$

$$C_{ij} - \bar{P}_{ij} \geq C_{(i-1)j} \quad \forall i \in I, \forall j \in J \mid i > 1 \tag{5}$$

$$C_{ij} - \bar{P}_{ij} + M(1 - X_{j(r+1)}) \geq C_{ih} - M(1 - X_{hr}) \quad \forall i \in I, \forall j \in J, \forall h \in J \mid j \neq h, r < |R| \tag{6}$$

$$C_{max} \geq C_{ij} \quad \forall i \in I, \forall j \in J \tag{7}$$

$$C_{max} \geq 0 \tag{8}$$

$$C_{ij} \geq 0 \quad \forall i \in I, \forall j \in J \tag{9}$$

$$X_{jr} \in \{0,1\} \quad \forall j \in J, \forall r \in R \tag{10}$$

The objective function (1) corresponds to the minimization of the completion time of the last job, i.e the makespan. Constraints (2) and (3) guarantee that every job is assigned to one position in the sequence and in each position has only one job. Constraints (4) are related to the completion time of jobs for the first worker. Constraints (5) and (6) calculates the completion time of jobs for the remaining workers. Constraints (7) define the makespan. Constraints (8) and (9) are the classical non-negative constraints that are here not compulsory as regards the Constraints (4). Constraints (10) define the binary decision variables.

The flow-shop scheduling problem is known in the literature to be NP-hard for the case of three or more production resources with makespan minimization (Pinedo, 2018). This means that the computational time to obtain optimal solutions increases exponentially as the size of the instance increases. In addition, the flow-shop scheduling problem that minimizes the makespan with learning effect is also proven to NP-hard, even for the case of two resources (workers), so the Johnson's rule does not reach an optimal solution (Wang & Xia, 2005).

2.2 Models for processing times with learning effect

This paper considers that workers improve their productivity over time. This means that the actual processing time for the

execution of a job is computed as a function of a learning effect. We assume that all the workers have the same learning rate. For this study, four approaches to model the learning effect were considered and further numerically compared with the problem without learning. These approaches were selected from models reported in the literature, which are based either on the position or on the sum-of-processing-times. While the DeJong model is growing in strength (Glock et al. 2019), it is not included in this document as this model addresses other parameters of a different nature (such as prior experience which could be better estimated from a case study). Consequently, its performance could not be compared to those of other cases.

So, position-based approaches mean that the job processing times decrease with the number of repetitions rather than accumulated time, while sum-of-processing-time are based on the proposition that the more a worker executes a job, the more performance he/she gets. Realizing that the learning effect is not endless, a control parameter has been included in these cases.

The mathematical model presented before can be modified in order to take into account the different approaches for modelling the learning effect, as proposed by several authors such as Biskup (2008), or Azzouz et al. (2018). The proposed cases are described as below.

- *Case 0*: In the basic problem, \bar{P}_{ij} is the baseline processing time without learning, of the j -th operation of job i
- *Case 1*: with position-based learning $P_{ijr} = \bar{P}_{ij}r^\alpha$
- *Case 2*: with truncated position-based learning $P_{ijr} = \bar{P}_{ij} \max\{r^\alpha, \beta\}$
- *Case 3*: with sum-of-processing-time based learning $P_{ijr} = (1 + \theta \sum_{k=1}^{r-1} P_{ijk})^\alpha \bar{P}_{ij}$
- *Case 4*: with truncated sum-of-processing-time based learning $P_{ijr} = \max\{(1 + \theta \sum_{k=1}^{r-1} P_{ijk})^\alpha, \beta\} \bar{P}_{ij}$

Where α is the learning index ($\alpha < 0$), β is a control parameter with $0 < \beta < 1$, and θ is a conversion factor (e.g., 1/60 to convert minutes to seconds)

In *case 1* the job processing time is based on the position, so taking the baseline flow-shop model, a new decision variable is added:

P_{ij} : actual processing time of job j on worker i

To calculate the job processing time, constraints (11) and (12) are needed:

$$P_{ij} = \sum_{r \in R} \bar{P}_{ij} X_{jr} r^\alpha \quad \forall i \in I, \forall j \in J \quad (11)$$

$$P_{ij} \geq 0 \quad \forall i \in I, \forall j \in J \quad (12)$$

We replace the Constraints (4), (5) and (6) by (13), (14) and (15) respectively.

$$C_{1j} \geq P_{1j} \quad \forall j \in J \quad (13)$$

$$C_{ij} - P_{ij} \geq C_{(i-1)j} \quad \forall i \in I, \forall j \in J | i > 1 \quad (14)$$

$$C_{ij} - P_{ij} + M(1 - X_{j(r+1)}) \geq C_{ih} - M(1 - X_{hr}) \quad (15)$$

$$\forall i \in I, \forall j \in J, \forall h \in J | j \neq h, r < |R|$$

From *case 1* and changing constraints (11) for (16), we would get the *case 2*.

$$P_{ij} = \sum_{r \in R} \bar{P}_{ij} X_{jr} \max\{r^\alpha, \beta\} \quad \forall i \in I, \forall j \in J \quad (16)$$

In *case 3*, the decision variables C_{ij} and P_{ij} were replaced by C_{ir} and P_{ir} .

C_{ir} : completion time of the job scheduled in the position the position $r \in R$ for the worker $i \in I$ m

P_{ir} : actual processing time of the job scheduled in the position $r \in R$ for the worker $i \in I$

Furthermore, equations (17) and (18) were included replacing the equation (11) from *case 1*.

$$P_{ir} = \left(1 + \theta \sum_{q \in R, q < r} P_{iq}\right)^\alpha \left(\sum_{j \in J} \bar{P}_{ij} X_{jr}\right) \quad (17)$$

$$\forall i \in I, \forall r \in R | r > 1$$

$$P_{i1} = \left(\sum_{j \in J} \bar{P}_{ij} X_{j1}\right) \quad \forall i \in I \quad (18)$$

$$C_{ir} \geq 0 \quad \forall i \in I, \forall r \in R \quad (19)$$

Constraint (19) is the conventional non-negative constraint. However, it is redundant as regards the constraint (20), (21) and (22).

$$C_{1r} \geq P_{1r} \quad \forall r \in R \quad (20)$$

$$C_{ir} - P_{ir} \geq C_{(i-1)r} \quad \forall i \in I, \forall r \in R | i > 1 \quad (21)$$

$$C_{ir} - P_{ir} + M(1 - X_{j(r+1)}) \geq C_{ih} - M(1 - X_{hr}) \quad (22)$$

$$\forall i \in I, \forall j \in J, \forall h \in J | j \neq h, r < |R|$$

Constraints (8) and (9) are the conventional non-negative constraints, but they become redundant because of the set of Constraints (4).

For *case 4*, the set E : states $\{1: Learning 2: Truncate\}$ was taken into consideration, as were the two variables which are defined:

U_{ire} : actual processing time of the job scheduled in position $r \in R$ for the worker $i \in I$ in the state $e \in E$

ξ_{ir} : maximum processing time of the job scheduled in position $r \in R$ for the worker $i \in I$

Equations (23), (24), (25), (26) and (27) are used to calculate the job processing time instead of equations (17) and (18). Constraint (28) and (29) are the non-negative constraint.

$$P_{i1} = \left(\sum_{j \in J} X_{j1} \right) \quad \forall i \in I \quad (23)$$

$$U_{ir1} = \left(1 + \theta \sum_{q \in R} P_{iq} \right)^\alpha \left(\sum_{j \in J} \bar{P}_{ij} X_{jr} \right) \quad \forall i \in I, \forall r \in R | r > 1 \quad (24)$$

$$U_{ir2} = \sum_{j \in J} \bar{P}_{ij} X_{jr} \beta \quad \forall i \in I, \forall r \in R | r > 1 \quad (25)$$

$$\xi_{ir} \geq U_{ire} \quad \forall i \in I, \forall r \in R, \forall e \in E \quad (26)$$

$$P_{ir} = \xi_{ir} \quad \forall i \in I, \forall r \in R | r > 1 \quad (27)$$

$$U_{ire} \geq 0 \quad \forall i \in I, \forall r \in R, e \in E \quad (28)$$

$$\xi_{ir} \geq 0 \quad \forall i \in I, \forall r \in R \quad (29)$$

We get three MILP models (*case 0*, *case 1* and *case 2*) and the two mixed-integer non-linear programming (MINLP) models (*case 3* and *case 4*).

3 COMPUTATIONAL EXPERIMENTS AND RESULTS

3.1 Experimental setting

The mathematical models were encoded on Python and solved using Pyomo (Hart et al., 2011). In order to evaluate and compare the suggested model for each case, several experiments were designed. These experiments included the number of machines (workers), jobs, learning and control parameters.

Integer values of processing times were generated using a uniform distribution between 1 and 100. The number of

workers was 2, while the number of jobs was defined to be 5 and 7. When required, values of α were fixed to be -0.152 , -0.322 , and -0.515 , and the values of β were fixed as 0.25 , 0.5 , and 0.75 . The rationale for the values of these two parameters is based on the most common values used in the literature (e.g., Amirian & Sahraeian, 2015; Rudek & Rudek, 2013; Wu et al., 2020). A total of 30 instances for each combination of jobs-machines were executed, giving a set of 1500 experiments.

Experiments were carried out on a PC with AMD Ryzen 2.0GHz and RAM 8 Gb. Glpk and Bonmin solvers (Hart et al., 2011) were used for solving the linear (*case 0*, *case 1* and *case 2*) and non-linear cases (*case 3* and *case 4*) respectively.

3.2 Analysis of results

In order to illustrate the impact of the learning effect modelling on the schedule, Figure 1 presents the Gantt chart for an exemplary instance of five jobs and two workers, $\alpha = -0.515$ and $\beta = 0.5$ if required. The makespan reaches a variation of 43.8% with *case 1* from the baseline, whereas for *case 3* the variation is 42.2%.

As shown in Figure 1, position-based learning effect models (*case 1* and *case 2*) allow for a faster reduction in the objective function (makespan), as regards models based on the sum-of processing times (*case 3* and *case 4*). Furthermore, position-based models are the most common in the literature regarding scheduling problem with the learning effect (Glock et al., 2019). However, it is considered that human behavior is closer to the precept than the more you practice, the better performance you obtain (based on the sum-of processing time) (Azzouz et al., 2018).

Therefore, the results are relevant because they show that even if the same parameter values are maintained, the selected model may have a significant effect on the objective function.

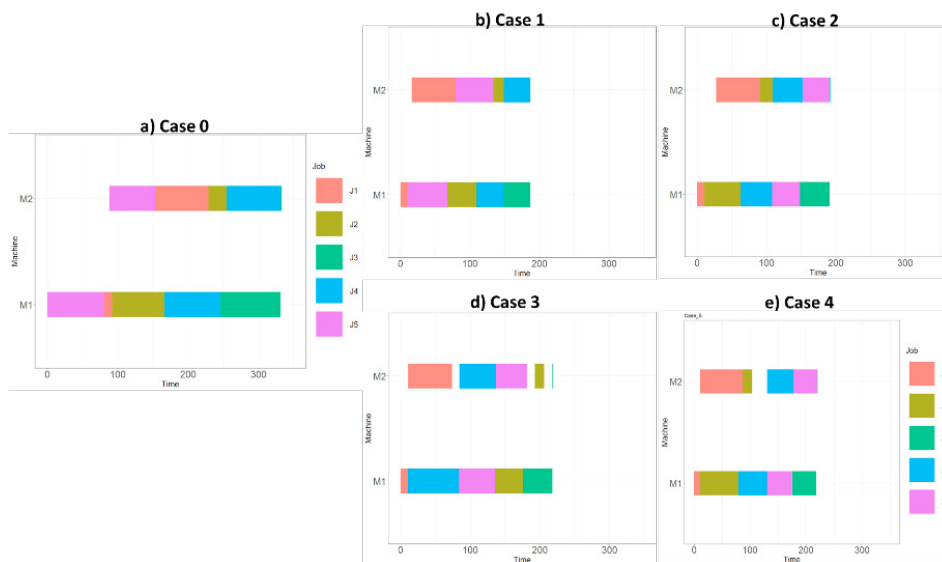


Figure 1. Gantt charts: a) Case 0, b) Case 1, c) Case 2, d) Case 3, e) Case 4

Regarding the outcomes of the full experiment (see Table 1), it is important to note that *case 1* and *case 2* that model the actual processing time based on the position, reduce the job

processing time faster than *case 3* and *case 4*. Similarly, it was found that in the case with a truncation parameter, the objective function is higher than those without a truncation

parameter. This is due to the control parameters that play an important role in the performance of the makespan. They represent the fact that learning cannot be regarded as infinite. However, these parameters need to be carefully evaluated. So, the minimum average deviation (for the 2 workers and 5 jobs problem and $\alpha = 0.152$) from the base line makespan (12.36%) is observed with *case 3* and *case 4*. It happens with *case 4* when truncation value is not reached due to low truncation value.

It can also be seen that the CPU time values change significantly between experiments with 5 and 7 jobs, especially those that are not linear, such as *case 3* and *case 4*. This fact supports the outcomes previously reported by Rudek (2011) about the complexity of the problems under study, showing that they belong to class of NP-hard problems. Therefore, the CPU time of *case 4* with 5 jobs is only 2.2% of the CPU time of the same case with 7 jobs. Thus, intending to test with 8 jobs for these cases, it was not possible to obtain the optimal solution after 8 hours of running time.

Table 1. Experimental results

Case	Workers	Jobs	Replications	α	β	Makespan		Average deviation from baseline makespan	CPU time (sec)		
						Average	Standard deviation		Average	Standard deviation	Min - Max
Case 0	2	5	30	-	-	306.37	52.81	-	0.13	0.01	0.12 - 0.15
Case 1	2	5	30	-0.152	-	262.45	45.36	-14.33%	0.16	0.01	0.14 - 0.19
Case 1	2	5	30	-0.322	-	223.12	39.46	-27.17%	0.16	0.01	0.13 - 0.18
Case 1	2	5	30	-0.515	-	209.22	29.48	-31.71%	19.47	3.75	14.61 - 30.7
Case 2	2	5	30	-0.152	0.25	262.45	45.36	-14.33%	0.15	0.02	0.13 - 0.23
Case 2	2	5	30	-0.152	0.5	262.45	45.36	-14.33%	0.15	0.01	0.13 - 0.17
Case 2	2	5	30	-0.152	0.75	262.45	45.36	-14.33%	0.14	0.01	0.13 - 0.17
Case 2	2	5	30	-0.322	0.25	223.12	39.46	-27.17%	0.15	0.01	0.13 - 0.16
Case 2	2	5	30	-0.322	0.5	223.12	39.46	-27.17%	0.14	0.01	0.13 - 0.17
Case 2	2	5	30	-0.322	0.75	243.65	42.28	-20.47%	0.15	0.01	0.13 - 0.17
Case 2	2	5	30	-0.515	0.25	187.84	33.6	-38.69%	0.14	0.01	0.13 - 0.17
Case 2	2	5	30	-0.515	0.5	192.29	34.34	-37.24%	0.14	0.01	0.13 - 0.17
Case 2	2	5	30	-0.515	0.75	240.82	41.83	-21.40%	0.14	0.01	0.13 - 0.17
Case 3	2	5	30	-0.152	-	268.58	42.84	-12.36%	18.56	2.81	15.03 - 27.57
Case 3	2	5	30	-0.322	-	236.88	35.44	-22.68%	19.5	4.79	13.87 - 33.75
Case 3	2	5	30	-0.515	-	209.22	29.48	-31.71%	19.47	3.75	14.61 - 30.7
Case 4	2	5	30	-0.152	0.25	268.5	42.84	-12.36%	21.48	3.58	17.15 - 30.3
Case 4	2	5	30	-0.152	0.5	268.5	42.84	-12.36%	20.84	3.04	16.48 - 28.79
Case 4	2	5	30	-0.152	0.75	268.5	42.84	-12.36%	22.16	3.87	17.37 - 31.99
Case 4	2	5	30	-0.322	0.25	236.58	35.37	-22.78%	22.94	3.78	17.97 - 35.72
Case 4	2	5	30	-0.322	0.5	236.58	35.37	-22.78%	24.15	6.79	16.92 - 38.53
Case 4	2	5	30	-0.322	0.75	249.39	40.9	-18.60%	24	4.51	16.81 - 39.53
Case 4	2	5	30	-0.515	0.25	208.69	29.3	-31.88%	23.66	4.49	17.55 - 32.93
Case 4	2	5	30	-0.515	0.5	209.2	29.85	-31.71%	22.42	2.84	16.33 - 29.8
Case 4	2	5	30	-0.515	0.75	245.67	41.43	-19.81%	25.62	7.82	17.14 - 50.77
Case 0	2	7	30	-	-	412.13	60.91	-	1.87	0.46	1.36 - 3.55
Case 1	2	7	30	-0.152	-	338.03	51.98	-17.98%	2.34	0.11	1.91 - 2.56
Case 1	2	7	30	-0.322	-	272.75	43.43	-33.82%	2.23	0.13	1.91 - 2.51
Case 1	2	7	30	-0.515	-	216.7	36.39	-47.42%	2.03	0.16	1.59 - 2.35
Case 2	2	7	30	-0.152	0.25	338.03	51.98	-17.98%	2.4	0.34	1.91 - 4.09
Case 2	2	7	30	-0.152	0.5	338.03	51.98	-17.98%	2.33	0.1	1.91 - 2.46
Case 2	2	7	30	-0.152	0.75	338.48	52.05	-17.87%	2.38	0.27	2.22 - 3.78
Case 2	2	7	30	-0.322	0.25	272.75	43.43	-33.82%	2.28	0.3	1.92 - 3.73
Case 2	2	7	30	-0.322	0.5	272.75	43.43	-33.82%	2.23	0.11	1.9 - 2.36
Case 2	2	7	30	-0.322	0.75	322.73	49.1	-21.69%	2.29	0.1	1.91 - 2.4
Case 2	2	7	30	-0.515	0.25	216.7	36.39	-47.42%	2.03	0.16	1.59 - 2.26
Case 2	2	7	30	-0.515	0.5	239.66	39.3	-41.85%	2.11	0.14	1.8 - 2.34
Case 2	2	7	30	-0.515	0.75	320.48	48.44	-22.24%	2.36	0.32	1.89 - 3.99
Case 3	2	7	30	-0.152	-	345.33	58.66	-16.21%	1545.6	82.59	1298.52 - 1698.71
Case 3	2	7	30	-0.322	-	295.08	45.83	-28.40%	1568.47	95.38	1217.26 - 1728.54
Case 3	2	7	30	-0.515	-	253.19	36.14	-38.57%	1527.75	126.08	1234.86 - 1770.29
Case 4	2	7	30	-0.152	0.25	345.14	58.66	-16.26%	1780.96	83.64	1601.88 - 2001.2
Case 4	2	7	30	-0.152	0.5	345.14	58.66	-16.26%	1788.59	75.49	1664.55 - 1976.39
Case 4	2	7	30	-0.152	0.75	345.22	58.81	-16.24%	1766.17	102.48	1576.24 - 2045.67
Case 4	2	7	30	-0.322	0.25	294.5	45.9	-28.54%	1766.17	102.48	1576.24 - 2045.67
Case 4	2	7	30	-0.322	0.5	294.5	45.9	-28.54%	1766.6	99.53	1606.46 - 2045.38
Case 4	2	7	30	-0.322	0.75	323.7	57.9	-21.46%	1769.39	98.74	1568.32 - 1966.54
Case 4	2	7	30	-0.515	0.25	252.42	36.18	-38.75%	1708.86	121.19	1508.64 - 2032.83
Case 4	2	7	30	-0.515	0.5	256.75	39.6	-37.70%	1738.64	134.08	1465.45 - 2090.44
Case 4	2	7	30	-0.515	0.75	319.68	57.79	-22.43%	1714.05	138.6	1512.3 - 2295.98

4 CONCLUSIONS AND PERSPECTIVES

This paper studied the flow-shop scheduling problem with learning effect in processing times. Several modelling approaches of the learning effect were considered leading to linear and non-linear mathematical formulations. The case of two workers was evaluated, and because of its NP-hardness, computational experiments were carried out with 5 and 7 jobs. The results of this study show that the way to model the learning effect in the flow-shop scheduling problem has an impact on the objective function (the makespan in this case), and the CPU time required to achieve the optimal solution. Thus, applying the approaches to cases that represent human behavior in a better way (*case 3* or *case 4*), imply greater CPU time. It is expected to cover other learning models, such as models which combine the principle of position and the sum-of-processing-times or the DeJong model. If necessary, the models mentioned will serve as a basis for the development of a new model, involving key factors in the learning process such as the previous experience and operational interruptions.

As well as, because of the complexity of this type of model (Chen et al., 2017 ; Rudek, 2011 ; Rudek & Rudek, 2013) and supported by the CPU time resulting from the current experiments, alternative methods should be proposed to get near-optimal solutions within a reasonable time; for instance, the development of greedy heuristics that have outstanding performance to solve the flow-shop scheduling problem (Rudek 2021). Although this paper deals with the problem at the theoretical level, the parameters and methodology could be validated from practical data (Grosse & Glock, 2015). Similarly, the approach of a case study will allow to represent real situations, to identify other types of needs existing in business, and to become useful decision tools for managers (Glock et al. 2019).

ACKNOWLEDGEMENTS

We thank Universidad de La Sabana for its financial support (project: INGPLD-45-2021) and the Embassy of France in Colombia that contribute with a mobility grant.

REFERENCES

- Amirian, H., & Sahraeian, R. (2015). Augmented ϵ -constraint method in multi-objective flowshop problem with past sequence set-up times and a modified learning effect. *International Journal of Production Research*, 53(19), 5962–5976.
- Azzouz, A., Ennigrou, M., & Ben Said, L. (2018). Scheduling problems under learning effects: classification and cartography. *International Journal of Production Research*, 56(4), 1642–1661.
- Biskup, D. (2008). A state-of-the-art review on scheduling with learning effects. *European Journal of Operational Research*, 188(2), 315–329.
- Bentefouet, F., & Nembhard, D. A. (2013). Optimal flow-line conditions with worker variability. *International Journal of Production Economics*, 141(2), 675–684.
- Chen, X., Chau, V., Xie, P., Sterna, M., & Błażewicz, J. (2017). Complexity of late work minimization in flow shop systems and a particle swarm optimization algorithm for learning effect. *Computers and Industrial Engineering*, 111, 176–182.
- Cheng, T. C. E., & Wang, G. (2000). Single Machine Scheduling with Learning Effect Considerations. *Annals of Operations Research*, 98(1–4), 273–290.
- 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 and Industrial Engineering*, 131(October 2018), 422–441.
- Grosse, E. H., & Glock, C. H. (2015). The effect of worker learning on manual order picking processes. *International Journal of Production Economics*, 170, 882–890.
- Hart, W. E., Watson, J. P., & Woodruff, D. L. (2011). Pyomo: Modeling and solving mathematical programs in Python. *Mathematical Programming Computation*, 3(3), 219–260.
- Kuo, W. H., & Yang, D. L. (2006). Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect. *European Journal of Operational Research*, 174(2), 1184–1190.
- Pinedo, M. L. (2012). *Scheduling: Theory, Algorithms, and Systems*. In Springer (Vol. 1, Issue 1).
- Rudek, A., & Rudek, R. (2013). Makespan minimization flowshop with position dependent job processing times - Computational complexity and solution algorithms. *Computers and Operations Research*, 40(8), 2071–2082.
- Rudek, R. (2011). Computational complexity and solution algorithms for flowshop scheduling problems with the learning effect. *Computers and Industrial Engineering*, 61(1), 20–31.
- Wang, J. B., & Xia, Z. Q. (2005). Flow-shop scheduling with a learning effect. *Journal of the Operational Research Society*, 56(11), 1325–1330.
- Wu, C.-C., Bai, D., Azzouz, A., Chung, I.-H., Cheng, S.-R., Jhwueng, D.-C., Lin, W.-C., & Said, L. B. (2020). A branch-and-bound algorithm and four metaheuristics for minimizing total completion time for a two-stage assembly flow-shop scheduling problem with learning consideration. *Engineering Optimization*, 52(6), 1009–1036.