# Comparison of two linear modeling of the learning effect in a two-resources flowshop

Yenny Alexandra Paredes-Astudillo[1 2], Valérie Botta-Genoulaz[1], Jairo R. Montoya-Torres[2]

[1] Univ Lyon, INSA Lyon, Université Claude Bernard Lyon 1, Univ Lumière Lyon 2, DISP, EA4570, 69621 Villeurbanne, France
{yenny.paredes-astudillo, valerie.botta}@insa-lyon.fr

[2] School of Engineering, Universidad de La Sabana, km 7 autopista norte de Bogotá D.C., Chía, Colombia
{yennypaas, jairo.montoya}@unisabana.edu.co

**Keywords**: *Scheduling, flowshop, learning effect, mathematical modeling.*

## 1 Introduction

Despite the technological advances in productive systems, the inclusion of workers is essential as well as the development of models that approach human behavior. In recent years, scheduling problems have started to integrate human factors. However, research opportunities exist to propose realistic models taking these factors into account when calculating job processing times in hand-intensive manufacturing system. The aim of this work is to compare two approaches to modeling the learning effect proposed in the literature in order to evaluate their complexity and to assess the effect over the makespan in a flowshop scheduling problem (FSSP).

## 2 Problem description and proposed solution approach

We consider a flowshop configuration where the *M* resources are workers, and a set of *N* independent jobs are to be processed in order to minimize the makespan. Each worker can process one job at a given time and preemption of a job is not allowed (that is, the execution of a job cannot be interrupted once its processing has started). All workers are available at the beginning of the scheduling horizon and have a 100% production rate.

In the basic problem (named "case 1"), $\bar{p}_{ij}$ is the baseline processing time without learning of the *i*-th operation of job *j*. We then consider two different ways to model the learning effect, a position-based learning [1] (hereafter named "case 2"), and a truncated position-based learning [2] (hereafter named "case 3"), where the processing time of each operation *j* can then be calculated respectively as follow:

- Case 2: FSSP with position-based learning $\quad P_{ijr} = \bar{P}_{ij} r^{\alpha}$
- Case 3: FSSP with truncated position-based learning $\quad P_{ijr} = \bar{P}_{ij} * \max\{r^{\alpha}, \beta\}$

Where $r$ is the position of job $j$ in the sequence, $P_{ijr}$ is the actual processing time of the $i$-th operation of job $j$ located in position $r$, $\alpha$ is a parameter of learning effect ($\alpha<0$), and $\beta$ is a control parameter with $0<\beta<1$.

MILP-type models have been developed for all the three cases and numerical experiments were undertaken in order to evaluate the impact of the two kinds of learning effect modeling.

# 3 Results

MILP models were coded on Python and were solved using Pyomo library [3]. Processing times were generated using a uniform distribution between 1 and 100. In this preliminary experiments, 2 workers and 8 jobs were considered. The rationale for these values is based on the computational complexity of the flowshop problem, which increases the CPU time when increasing the number of jobs and workers. When required, the values of $\alpha$ were fixed to be – 0.152, – 0.322, and – 0.515, and the values of $\beta$ were fixed as 0.25, 0.50, and 0.75. A total of 30 replications for each combination were performed. Table 1 shows the average results obtained. We can conclude that the variation of makespan depends on the learning rate, so a lower value of $\alpha$ means a greater variation of makespan relative to the baseline scenario. However, lower $\alpha$ values and higher $\beta$ values (case 3) reduce the variation.

| Case | Workers | Jobs | Replications | α | β | Makespan | | Average deviation from baseline makespan | CPU time (sec) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Average | Standard deviation | | Average | Standard deviation | Min-Max |
| Case 1 | 2 | 8 | 30 | - | - | 459.50 | 65.27 | - | 53.83 | 14.50 | 34.49 -113.23 |
| Case 2 | 2 | 8 | 30 | -0.152 | - | 369.14 | 53.95 | -19.70% | 75.34 | 12.30 | 62.98 -131.08 |
| Case 2 | 2 | 8 | 30 | -0.322 | - | 292.75 | 44.39 | -36.34% | 61.89 | 7.38 | 44.30 -83.41 |
| Case 2 | 2 | 8 | 30 | -0.515 | - | 227.16 | 36.65 | -50.63% | 45.27 | 8.71 | 20.97 -69.14 |
| Case 3 | 2 | 8 | 30 | -0.152 | 0.25 | 369.14 | 53.95 | -19.70% | 75.70 | 11.32 | 63.27 -119.55 |
| Case 3 | 2 | 8 | 30 | -0.152 | 0.50 | 369.14 | 53.95 | -19.70% | 75.29 | 11.51 | 63.61 -126.32 |
| Case 3 | 2 | 8 | 30 | -0.152 | 0.75 | 370.93 | 54.08 | -19.30% | 74.90 | 10.43 | 58.67 -106.95 |
| Case 3 | 2 | 8 | 30 | -0.322 | 0.25 | 292.75 | 44.39 | -36.34% | 62.71 | 7.57 | 43.68 -85.30 |
| Case 3 | 2 | 8 | 30 | -0.322 | 0.50 | 295.75 | 44.39 | -36.34% | 62.64 | 7.94 | 43.96 -87.03 |
| Case 3 | 2 | 8 | 30 | -0.322 | 0.75 | 355.91 | 51.79 | -22.56% | 74.37 | 8.78 | 59.53 -90.50 |
| Case 3 | 2 | 8 | 30 | -0.515 | 0.25 | 227.16 | 36.65 | -50.63% | 44.76 | 8.38 | 21.29 -65.16 |
| Case 3 | 2 | 8 | 30 | -0.515 | 0.50 | 260.20 | 40.10 | -43.41% | 51.86 | 10.62 | 30.28 -76.38 |
| Case 3 | 2 | 8 | 30 | -0.515 | 0.75 | 353.76 | 51.44 | -23.03% | 74.42 | 7.34 | 61.67 -89.74 |

TAB 1 – Experimental results

# 4 Conclusions and perspectives

The originality of the study lies in the fact that it reviews different approaches proposed in the literature to model the learning effect in the FSSP, experimentally evaluate the impact of such learning effect modeling approaches. We will continue to work to overcome the current limitations (e.g., only linear learning effect models, computing time). Other modelling approaches could also be included, and given the complexity of these type of problems, alternative solution methods will be required. In addition, the deteriorating effect is expected to be incorporated.

# References


[1] Biskup, D. (2008). A state-of-the-art review on scheduling with learning effects. *European Journal of Operational Research, 188(2), 315-329.*

[2] Cheng, T. C. E., Wu, C. C., Chen, J. C., Wu, W. H., & Cheng, S. R. (2013). Two-machine flowshop scheduling with a truncated learning function to minimize the makespan. *International Journal of Production Economics*, 141(1), 79-86..

[3] 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.