Available online at http://scik.org
J. Math. Comput. Sci. 10 (2020), No. 5, 2033-2052
https://doi.org/10.28919/jmcs/4789
ISSN: 1927-5307

SIMULTANEOUS CONSIDERATION OF SEQUENCE-DEPENDENT AND POSITION-DEPENDENT SETUP TIMES IN SINGLE MACHINE SCHEDULING PROBLEMS

FRANCISCO ANGEL-BELLO[1,*], JOBISH VALLIKAVUNGAL[1], ADA ALVAREZ[2]

[1]Tecnologico de Monterrey, Escuela de Ingenieria y Ciencias, Mexico

[2]Universidad Autónoma de Nuevo León, Facultad de Ingenieria Mecánica y Eléctrica, Mexico

Abstract. In this work we address a single machine scheduling problems with sequence-dependent setup times, in which the setup time and the processing time may depend on the job position in the processing order. We consider two manufacturing environments. In the first one, jobs are processed automatically, then the job positions affect only on the setup times. In the second environment, the operators modify the machine settings between different types of jobs and operate it to process the jobs, then the job positions affect both setup time and processing time. In this work, we will investigate the validity of the assumption: scheduling problems render high-quality solutions that reckon with job positions just as the those reckoned without job positions. Minimising the maximum completion time and the sum of completion time of the jobs are the objective functions. To tackle this scenario, we introduce four mathematical formulations: one formulation for each combination of objective function and manufacturing environment. The validity of the models is established by the results of extensive computational experiments on the proposed models.

Keywords: sequence-dependent setups; position-dependent setups; single machine scheduling; integer formulations; makespan; total completion time.

2010 AMS Subject Classification: 90B30, 90B35.

*Corresponding author

E-mail address: fangel@tec.mx

Received June 23, 2020

1. INTRODUCTION

There is an immense time-saving when setup times (costs) have been explicitly included in scheduling decisions in various real world industrial and service realms. It is crucial, in some cases, that the setup times must be explicitly considered. Food processing, chemical, printing or metal processing industries just to name a few [2]. Whenever the time required to prepare a machine for processing a given job is also depends on the last scheduled job, the setup time is called as sequence-dependent setup time (ST_{sd}).

The explicit consideration of ST_{sd} in scheduling problems substantially increases the complexity and hence the problems will be much harder to solve or have to approximate from a computational point of view. For instance, the problem of minimising the Total Completion Time (TCT) on a machine with independent setup times can be easily solved by the simple shortest processing time rule, whereas if ST_{sd} are considered, the problem becomes an asymmetric minimum latency problem, which is NP-hard in the strong sense [9].

The dependency can be position-dependent, which means that the time required to perform the operations vary depending on the job positions in the processing sequence. When the time variations are a consequence of operation repetitions, they are known as learning effect or deterioration effect in the scheduling literature. If it decreases as the job position grows then it is possible to consider that there is a learning effect, while if it increases it is possible to consider that there is a deterioration effect. Hence, the variation in the operation times due to job positions is regarded as either of these effects.

Many researchers have been studied extensively the deterioration and learning effects. In the literature have found only the problems in which it is assumed that there are no setup times or that they depend only on the job that is going to be processed that can be included in the job processing times. De facto, the studies have been carried out only on the processing times of the jobs. Having said that, we address single machine scheduling problems with ST_{sd}, where both, setup times and processing times, may be affected by position-dependent learning effects.

We consider two manufacturing environments. In the first one, the jobs are processed automatically, while the machine settings between different types of jobs are executed manually (the position affects only the setup times). In the second one, the operators modify the machine settings between jobs and operate the machine to process the jobs (the position affects both setup and processing times). Two performance measures, maximum completion time (makespan, C_{max}) and total completion time (TCT), are used. The former is focused on the efficient use of machines and the latter focused on the maximisation of the production flow and the minimisation of the work-in-process inventories.

In this work, we will validate the assumption that prime solutions of the problems that reckoned without job position effect on ST_{sd} are also prime solutions for the problems reckon with the effect. It is less obscure to obtain quality solutions when the job position effect is ignored. Then, if the assumption were true we could take those solutions and simply evaluate them using the corresponding job position factor to obtain the true values of the objective functions. Otherwise, it would be necessary to consider job position effect in the production programming process to obtain high-quality solutions.

This study will get through the decision makers to decide whether or not to include this kind of learning effect in the production planning to make more efficient the production process. For, we propose four mathematical formulations, one formulation for each combination of manufacturing environment and objective function. The effectiveness of the models are verified by carrying out extensive computational experiments on the models.

To the best of our knowledge, this is the first time that:

- mathematical formulations have been proposed for scheduling problems where setup times simultaneously depend on the sequence and on the positions of the jobs in the processing order.
- investigate how the quality of the solutions is affected by the positions of the jobs in the processing order when setup times are sequence-dependent.
- both ST_{sd} and processing time are affected by the positions of the jobs in the processing order.

In the next section, we will discuss the related literature. In section 3 the mathematical formulations of the problem are presented, then the computational tests followed by the conclusions.

2. LITERATURE REVIEW

The seminal work of learning effect in scheduling problems was presented by [6], hereinafter a myriad of works have been published about the learning effects on the processing times in scheduling problems. The approaches to tackle the problem have been distinguished [20] as: time-dependent [10], position-dependent [7] and cumulative [14].

In the time-dependent approach, the time needed to produce a unit decrease depends on the starting time of the job. [10] presents a comprehensive description of scheduling models with this effect. While, in the position-dependent approach, the time required to produce a unit decrease rely on the number of repetitions of jobs [6]. Here, processing time p_{jr} of job j in position r is calculated by: $p_{jr} = p_j \cdot f(a,r)$, where p_j is the processing time without learning effect (normal processing time), a is a constant learning factor and f is a decreasing function with respect to r.

In the cumulative approach, the time required to produce a unit decrease is depending on the normal processing time p_j and on an accumulated value of a parameter. Typically, it decreases depending on the sum of the processing times of the all jobs already scheduled [14]. Here, processing time p_{jr} is calculated by: $p_{jr} = p_j \cdot f\left(a, \sum_{k=1}^{r-1} p_{[k]}\right)$, where $p_{[k]}$ is the normal processing time of job in position k, and f is inversely proportional to $\sum_{k=1}^{r-1} p_{[k]}$. This effect is also known as the time-dependent learning effect or sum-of-processing-times-based learning effect.

Numerous studies use these approaches to investigate the effect of learning on various scheduling problems. Some recent works addressing single machine scheduling problems are [18, 8, 15, 13, 11, 23]. For more details about scheduling problems with learning effects see the surveys by [7, 19, 5].

A particular case of learning effect on processing times so-called past-sequence-dependent (p-s-d) setup time approach, is introduced by [12]. In p-s-d setup time approach the processing time p_{jr} is obtained as the normal processing time plus a value that depends on the sum of the processing times of the all already scheduled jobs; that is, $p_{jr} = s_{[r]} + p_j$, where $s_{[1]} = 0$, $s_{[r]} = b^{r-1} \cdot \sum_{k=1}^{r-1} p_{[k]}$, for $r = 1,2,3,\cdots,n$ and b is a constant. The $s_{[r]}$ is interpreted as a setup time that depends upon the sum of the processing times of the all already scheduled jobs. [24, 21, 22] used p-s-d approach and [1] presented a comprehensive survey on scheduling problems with p-s-d setup times.

All the works does not consider setup times for the machines or they depend only on the job to be processed and therefore they can be included in the job processing times. However, in practical industrial scenario, it is indispensable to consider the setup time explicitly. Despite the growing interest in scheduling problems involving sequence-dependent setup times, we only found one paper studying learning effect with this kind of setup times in single-machine scheduling problems. Recently, [16] addressed a scheduling problem on a single machine with ST_{sd}. They considered a position-dependent learning effect only on processing times and their goal was to minimise total tardiness.

In this paper, we address the minimisation of the C_{max} and the TCT in single machine scheduling problems with sequence-dependent setup times. We analyse position-dependent learning effects in two contexts: when the job positions affect only the setup times, and when they affect both setup times and processing times. The main objective is to investigate when it is necessary to consider the job positions in the production programming process to acquire finest solutions.

3. Formulation of the Problems

We consider a set of n independent jobs to be processed on a single machine. Each job j has a processing time p_j and there is a machine setup time s_{ij}, which is incurred when job j immediately follows job i. In general, $s_{ij} \neq s_{ji}$. At the beginning, the machine is at an initial state 0 (or dummy job 0), and there are setup time s_{0j}, prior to process the first job in the machine. All the jobs are available initially, there is an ST_{sd} between jobs, and the preemption is not allowed. The objective function is the minimisation of the C_{max} or the TCT for each manufacturing environment, considering position-dependent learning effects.

A learning effect is represented, in general, by the function $y = f(b,r)$, which depends on the number of r (setups) that have already been done in the machine and on a parameter b (learning factor) associated with the learning rate. In the first manufacturing environment, we assume that the position affects only the setup times. Then the total time, t^r_{ij}, required for processing the job j in position r just after job i in position $r-1$ is defined as:

$$t_{ij}^{r} = s_{ij}^{r} + p_j = f(b,r) \cdot s_{ij} + p_j \tag{1}$$

where s_{ij}^{r} denotes the machine setup time from job i in position $r-1$ to job j in position r.

In the second manufacturing environment, we assume that both the setup times and the processing times are affected. Then the total time, t_{ij}^{r}, required to process the job j in position r just after job i in position $r-1$ is defined as:

$$t_{ij}^{r} = s_{ij}^{r} + p_j^{r} = f(b,r) \cdot (s_{ij} + p_j) \tag{2}$$

where p_j^{r} denotes the processing time of job j in position r. The function $f(b,r) = b^{r-1}$, $b \in (0,1]$, represents the learning rate that is inversely proportional to the learning factor b.

In the four formulations, the objective is to find a sequence of jobs P that minimises the corresponding objective function (C_{max} or TCT). In order to evaluate the objective functions, the solutions are represented by a sequence of n jobs;

$$P = \left\{ 0, [1], [2], \cdots, [r-1], [r], [r+1], \cdots, [n] \right\} \tag{3}$$

where, $[r]$ denotes the job in the position r in the sequence P.

For a given sequence P, the *makespan* considering learning effect can be calculated by:

$$C_{max}^{LE}(P) = t_{0[1]}^{1} + t_{[1][2]}^{2} + \cdots + t_{[n-2][n-1]}^{n-1} + t_{[n-1][n]}^{n} \tag{4}$$

However, the TCT for a sequence P without learning effect can be calculated by [3]:

$$TCT(P) = nt_{0[1]} + (n-1)t_{[1][2]} + \cdots + 2t_{[n-2][n-1]} + t_{[n-1][n]} \tag{5}$$

where, $t_{[i][j]} = s_{[i][j]} + p_{[j]}$. Further, when the learning effect is considered, it can be calculated by:

$$TCT^{LE}(P) = nt_{0[1]}^{1} + (n-1)t_{[1][2]}^{2} + \cdots + 2t_{[n-2][n-1]}^{n-1} + t_{[n-1][n]}^{n} \tag{6}$$

In expressions (4) and (6) the contribution of each job to the objective function depends on its position in the sequence. Consequently, we should define decision variables that consider the position of the jobs in the sequence. Moreover, the time-dependent formulations for the minimum latency problem [3] can be adapted to this problem.

To formulate the mathematical models, we define the decision variables x_{ij}^r as:

$$x_{ij}^r = \begin{cases} 1, & \text{if and only if jobs } i \text{ and } j \text{ occupy positions } r \text{ and } (r+1) \text{ in the sequence} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Note that $x_{ij}^r = 1$ means that there are $(n-r)$ jobs after job i in the sequence. Using these variables, it is possible to define the objective functions with learning effect as follows. For the makespan:

$$\min C_{max}^{LE} = \sum_{i=1}^{n} t_{0i}^1 \sum_{\substack{j=1 \\ j\neq i}}^{n} x_{ij}^1 + \sum_{r=1}^{n-1} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i}}^{n} t_{ij}^{r+1} x_{ij}^r \tag{8}$$

and for the total completion time:

$$\min TCT^{LE} = n \sum_{i=1}^{n} t_{0i}^1 \sum_{\substack{j=1 \\ j\neq i}}^{n} x_{ij}^1 + + \sum_{r=1}^{n-1} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i}}^{n} (n-r) t_{ij}^{r+1} x_{ij}^r \tag{9}$$

The values of t_{ij}^r are calculated by (1) or (2) according to the manufacturing environment under consideration.

Note that, variables x_{ij}^1 appear in these two expressions, (8) and (9), of the objective functions. In the first expression, the variables x_{ij}^1 are used to calculate the contribution of the job in the first position of the processing order. While in (9) they are used to calculate the contribution of the job in the second position.

The set of constraints is defined as follows:

$$\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i}}^{n} x_{ij}^1 = 1 \tag{10}$$

$$\sum_{\substack{l=1 \\ l\neq i}}^{n} x_{li}^{r-1} - \sum_{\substack{j=1 \\ j\neq i}}^{n} x_{ij}^r = 0; \qquad (i = 1, 2 \ldots, n \quad r = 2, 3, \ldots, n-1) \tag{11}$$

$$\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i}}^{n} x_{ji}^{n-1} = 1 \tag{12}$$

$$\sum_{\substack{j=1\\j\neq i}}^{n} x^1_{ij} + \sum_{r=1}^{n-1}\sum_{\substack{i=1\\i\neq j}}^{n} x^r_{ji} = 1 \qquad (i = 1,2\ldots,n) \tag{13}$$

$$x^r_{ij} \in \{0,1\} \qquad (i,j = 1,2,\ldots,n;\, j \neq i;\; r = 1,2,\ldots,n-1) \tag{14}$$

Constraints (10) guarantee that a single job occupies position 1 in the sequence, while constraints (12) guarantee that a single job occupies the last position in the sequence. The flow conservation constraints (11) establish the sequence continuity. They state that a job i in position r of the sequence can have a successor in position $(r+1)$ if and only if it has a predecessor in position $(r-1)$. Constraints (13) assure each job to occupy a single position in the sequence. Constraints (14) establish the binary nature of the variables x^r_{ij}.

The studied problems can be seen as particular cases of time-dependent Asymmetric Travelling Salesman Problems (ATSPs). In addition, for $b = 1$, the problems for minimising the C_{max} can be transformed into ATSPs and the problems for minimising the TCT can be transformed into Asymmetric Minimum Latency Problems (AMLPs). Both the ATSP and the AMLP are NP-hard problems in a strong sense [17, 9].

Considering the objective functions and the manufacturing environments, we obtain 4 integer linear formulations that share the set of constraints (10)-(14). We will refer to these models according to the nomenclature presented in Table 1.

Table 1: Characteristics of the proposed models

	C_{max}	TCT
$t^r_{ij} = b^{r-1}s_{ij} + p_j$	Model I	Model III
$t^r_{ij} = b^{r-1}(s_{ij} + p_j)$	Model II	Model IV

The proposed models will be used with different set of instances to study the impact of the learning effects on solution quality when the setup times are sequence-dependent.

4. Computational Experiments

To analyse the impact of the position-dependent learning effect, we conducted two types of experiments. First, we tested how the different levels of the learning factor b affect the processing job's order. Second, we analyse how good the solution is, ignoring learning effects, when it is evaluated for different values of b. All experiments are performed in a system with Intel Core 2 Duo CPU at 3.00 GHz and 3.21 GB of RAM under Windows OS. The formulations are implemented in C++, and solved using concert technology of professional solver CPLEX 12.9.

A subset of instances generated by [4] with number of jobs $n = \{15, 20, 25, 30\}$, which is available at http://www.cima.uadec.mx/investigacion/instancias/, is used. The processing times p_j are generated using the uniform distribution in the interval [1-99], and the setup times are generated in three intervals: [1-49], [1-99] and [1-124], denoted by R_1, R_2, and R_3 respectively. Each class of instances contains 20 variants with each possible combination of number of jobs and range of setup times, that results in a total of 240 instances. First machine data has been taken since we are studying the single machine scenario. The values of b vary between 0.1 and 1.0 with a step of 0.1. We set $b = 1$ to obtain the optimal solutions without learning effect. The models are tested with each set of instances and for each value of b, which results in 9600 experiments. The results obtained for each objective is presented below.

4.1. Results for the makespan.

In this section we analyse the impact of the learning effect on the job processing order when the objective is to minimise the makespan.

In Table 2 we show the optimal sequences given by CPLEX using Model I and Model II on the instances with number of jobs 15 and for different levels of b . We can observe that different processing orders have been obtained for different learning levels. In the last column we show the quality of the optimal solution (obtained without learning effect) when it is implemented for the different levels of b. The gap for each value of b is calculated by:

$$gap(b)\% = \frac{SolVal_{withoutLE}(b) - OptimalVal(b)}{OptimalVal(b)} * 100 \tag{15}$$

where $OptimalVal(b)$ is the optimal solution value for a given value of b, and $SolVal_{withoutLE}(b)$ is the value of the solution without learning effect that is evaluated for the same b value.

Table 2: Optimal solutions obtained by Model I and Model II for C_{max} when number of jobs is 15

Model	b	optimal sequences															$gap(b)\%$
I	0.1	10	6	1	9	13	14	4	5	3	8	12	7	15	2	11	0.1171
	0.2	2	14	4	5	3	8	7	6	1	9	13	12	10	15	11	0.1689
	0.3	2	14	4	3	8	7	6	1	9	13	10	12	5	15	11	0.2861
	0.4	2	14	4	3	8	7	6	1	9	5	12	10	11	15	13	0.4342
	0.5	2	14	4	3	8	7	6	1	9	13	15	11	5	12	10	0.6191
	0.6	2	14	4	3	8	7	6	1	9	13	15	11	5	12	10	0.8406
	0.7	2	14	4	3	8	7	6	1	9	13	15	11	5	12	10	1.0735
	0.8	2	14	4	3	8	7	6	1	9	13	15	11	5	12	10	1.2133
	0.9	2	14	4	3	8	7	6	1	9	13	15	11	5	12	10	0.9550
	1.0	2	15	11	14	4	5	3	8	7	6	1	9	13	10	12	0.0000
II	0.1	4	1	6	12	9	15	11	14	5	3	13	10	8	2	7	335.2847
	0.2	6	4	1	9	12	11	14	5	7	10	15	8	3	13	2	298.3196
	0.3	6	4	1	9	15	12	11	14	5	3	7	10	8	13	2	275.7893
	0.4	6	4	1	9	15	12	11	14	5	3	7	8	10	2	13	245.8289
	0.5	6	1	9	4	7	14	12	10	15	11	5	3	8	2	13	212.0319
	0.6	6	1	9	4	7	14	12	10	15	11	5	3	8	2	13	172.7909
	0.7	6	1	9	4	7	14	12	10	15	11	5	3	8	2	13	122.5872
	0.8	6	1	9	4	7	14	12	10	15	11	5	3	8	2	13	67.3731
	0.9	1	9	4	7	6	12	11	14	5	3	8	10	2	15	13	23.0378
	1.0	2	15	11	14	4	5	3	8	7	6	1	9	13	10	12	0.0000

Observe in the last column that implementing the solution without learning effect ($b = 1$) for the first manufacturing environment with C_{max} as objective function has little influence over the makespan, whereas in the second manufacturing environment the opposite happens. In other words, using solutions obtained without considering learning effects lead us to implement mediocre sequences in the second manufacturing environment.

The behaviour observed in Table 2 is common for all tested instances as shown in Table 3. The first column in Table 3 displays the number of jobs, then the levels of the learning factor b. Columns corresponds to R_i gives the gaps for each setup range for Model I and Model II respectively. The gap values are calculated using the expression (15). Each class is a subset of 20 instances based on number of jobs (n), setup range (R) and learning factor (b) and the average value of each class is given.

Table 3: Gaps (%) of optimal solutions for C_{max} without learning effect regarding learning levels

		Model I			Model II		
n	b	R_1	R_2	R_3	R_1	R_2	R_3
15	0.1	0.2172	0.6450	0.7309	127.7368	91.4103	119.5835
	0.2	0.2494	0.7235	0.8188	129.0850	85.7219	107.6149
	0.3	0.2987	0.8245	0.9259	128.4908	78.8185	96.2292
	0.4	0.3629	0.9547	1.0624	124.6238	70.8520	85.5784
	0.5	0.4550	1.1166	1.2507	115.5320	61.8241	75.0923
	0.6	0.5797	1.3073	1.4955	100.1633	51.4803	63.2126
	0.7	0.7204	1.4723	1.7547	76.8231	38.8813	48.0305
	0.8	0.7921	1.3832	1.7743	46.5222	23.8488	29.0306
	0.9	0.5136	0.7797	1.0028	17.4165	8.8433	10.5954
20	0.1	0.1691	0.1779	0.4589	245.3804	70.2386	130.5053
	0.2	0.1923	0.2094	0.5165	231.9967	72.8259	124.7613
	0.3	0.2217	0.2531	0.5937	214.607	74.5855	118.4178
	0.4	0.2597	0.3121	0.6945	194.6465	74.6831	111.8462
	0.5	0.3115	0.3931	0.8407	171.9072	72.4727	103.2431
	0.6	0.3848	0.5117	1.0379	144.6652	67.0290	92.6586
	0.7	0.4810	0.6866	1.2877	110.9587	56.6477	76.5562

	0.8	0.5605	0.8386	1.4450	69.0802	38.9729	51.1544
	0.9	0.4316	0.6212	0.9745	26.4555	16.0844	20.0073
25	0.1	0.0955	0.1397	0.3906	250.7117	115.1427	127.6952
	0.2	0.1015	0.1699	0.4581	241.9626	117.4986	124.9340
	0.3	0.1133	0.2102	0.5373	233.6208	117.9886	120.9140
	0.4	0.1308	0.2623	0.6334	223.1144	116.8636	116.7138
	0.5	0.1585	0.3342	0.7591	207.5627	113.9188	110.3505
	0.6	0.2048	0.4433	0.9320	184.6238	107.1859	100.8466
	0.7	0.2802	0.6194	1.1866	150.8183	93.4388	86.1978
	0.8	0.3820	0.8364	1.5040	100.6101	67.6190	62.2399
	0.9	0.3549	0.7917	1.4260	41.2693	28.7953	27.0196
30	0.1	0.0630	0.2644	0.1412	169.4356	265.4605	145.0042
	0.2	0.0704	0.2728	0.1599	167.8989	249.3493	137.5331
	0.3	0.0827	0.2887	0.1868	167.9713	231.4849	133.1857
	0.4	0.0992	0.3119	0.2283	167.7002	210.2488	128.4251
	0.5	0.1249	0.3479	0.2936	164.4083	185.2614	122.0126
	0.6	0.1639	0.4068	0.4026	155.2849	156.9281	112.3899
	0.7	0.2259	0.5086	0.5826	136.7524	125.3460	97.4379
	0.8	0.3214	0.6816	0.8765	102.3536	87.2831	73.5631
	0.9	0.3614	0.7245	1.0642	46.7779	39.0982	34.7186

A graphical representation of behaviour of the gaps (on average) for each level of learning factor b and each setup range R is presented in Figure 1.

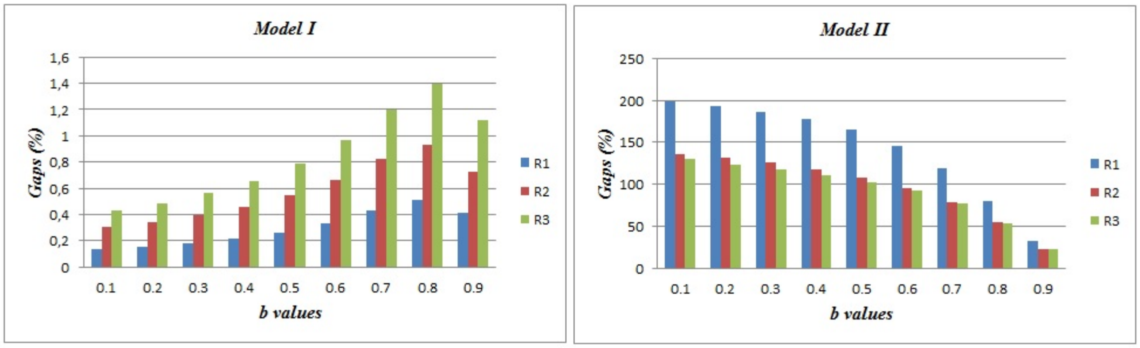

FIGURE 1. Average gaps for each setup time range and each b value.

It can be observed that in the first manufacturing environment the learning has very little effect on the production process duration, since the solution obtained without learning effect generally provides job sequences of good quality for all levels of the learning factor. Having said that, in the second manufacturing environment, we note that the sequence obtained without learning has low quality for the different levels of the learning factor. Therefore, in this case if we look for quality sequences it is quintessential to consider the learning effect in the production programming.

For the Model I (learning effect only on the setup times), the gaps grow along with the setup range, for each b. That is, the quality of the solution without learning effect is inferior when the range of variation of the setup times is greater than the range of variation of the processing times (R_3) for all levels of the learning factor. However, for Model II (learning effect on both the setup times and the processing times), the opposite happens. That is, the solutions without learning effect are degraded when the range of variation of the processing times is greater than the range of variation of the setup times (R_1). In addition, it can be observed that the gaps grow as the b values decrease, i.e., the gaps grow with the learning rate.

4.2. Results for the total completion time. Here we analyse the impact of the learning effect on the job order when the objective is to minimise the total completion time. Firstly, it is presented the results given by CPLEX solver for Model III and Model IV using instances with number of jobs 15.

Table 4: Optimal solutions obtained by Model III and Model IV for TCT using a 15-job instance

Model	*b*	optimal sequence															gap(*b*)%
III	0.1	2	1	11	9	13	6	4	12	5	8	7	3	10	15	14	15.6155
	0.2	2	1	11	9	13	6	4	12	5	8	7	3	10	15	14	14.0834
	0.3	2	1	11	9	13	6	4	12	5	8	7	3	10	15	14	12.2025
	0.4	2	1	11	9	13	4	6	12	5	8	7	3	10	15	14	9.9546
	0.5	2	5	1	11	9	13	4	6	12	8	7	3	10	15	14	8.4385
	0.6	2	5	1	11	9	13	4	6	12	8	7	3	10	15	14	6.4854
	0.7	2	5	1	8	9	11	13	4	12	6	7	3	10	15	14	4.6285
	0.8	2	5	1	8	9	11	13	4	12	3	6	7	10	15	14	2.9048
	0.9	2	5	1	8	9	11	13	4	12	3	6	7	14	10	15	0.1816
	1	2	5	6	1	8	9	13	4	12	3	11	10	15	7	14	0.0000
IV	0.1	2	1	11	9	13	4	12	8	6	7	5	10	15	3	14	10.0557
	0.2	2	1	11	9	13	4	12	8	6	7	5	15	3	14	10	15.9879
	0.3	2	1	11	9	13	4	12	8	6	7	5	15	3	14	10	18.2681
	0.4	2	1	8	9	11	5	6	12	4	15	13	3	14	7	10	18.5263
	0.5	2	1	8	9	11	5	6	12	4	15	13	3	14	7	10	17.8733
	0.6	2	1	8	9	11	5	6	12	4	15	13	3	14	7	10	15.7253
	0.7	2	1	8	9	11	5	6	12	4	15	7	14	13	3	10	11.9088
	0.8	2	1	8	9	11	5	12	4	15	6	7	14	13	3	10	6.6076
	0.9	2	1	8	9	11	5	12	4	15	6	7	14	13	3	10	1.0962
	1	2	5	6	1	8	9	13	4	12	3	11	10	15	7	14	0.0000

The last column gives the gaps of the optimal sequence ignoring the learning effect regarding optimal solutions obtained for each *b* value. These gap values are calculated using expression (15) for the *TCT* objective function. The gaps have similar pattern for both manufacturing environments and they increase as *b* values decrease. Thus, both environments can be affected in a similar tone by the learning effects. In order to refute or endorse this statement we show the results for all tested instances.

Table 5: Gaps (%) of optimal solutions for *TCT* without learning effect regarding learning levels

		Model I			Model II		
n	b	R_1	R_2	R_3	R_1	R_2	R_3
15	0.1	6.1103	11.3390	9.4763	20.3764	6.2276	13.9493
	0.2	5.7949	10.6910	8.8852	17.4676	7.1035	13.6800
	0.3	5.4789	9.9187	8.1433	15.4775	7.5417	12.6743
	0.4	5.1246	9.0211	7.1655	13.6905	7.6043	11.1685
	0.5	4.7052	8.0269	6.1133	11.2852	7.2802	9.3493
	0.6	4.0679	6.7245	4.9972	8.5842	6.4496	7.3038
	0.7	3.1224	4.9213	3.7362	5.9611	4.9298	5.0095
	0.8	2.0394	2.8022	2.0618	3.3743	2.8347	2.8082
	0.9	0.8367	0.8288	0.5609	1.0245	0.8850	0.6782
20	0.1	6.4371	8.8005	10.8290	55.7083	13.8469	21.2778
	0.2	6.1681	8.3507	10.306	48.1192	14.6689	19.6754
	0.3	5.8133	7.9582	9.7336	40.0923	15.0687	18.3611
	0.4	5.4159	7.5540	9.1023	32.5878	14.8299	17.2057
	0.5	5.0203	7.0093	8.2665	25.7075	13.8188	15.4862
	0.6	4.5118	6.1175	7.0491	19.1963	11.9541	13.4599
	0.7	3.7495	4.8691	5.5179	12.8249	9.14257	10.8345
	0.8	2.5826	3.1717	3.6202	6.9536	5.63898	6.8125
	0.9	1.0175	1.0446	1.3362	2.0585	1.7728	2.6426
25	0.1	4.7195	9.2083	10.9102	38.0189	21.1471	43.8776
	0.2	4.5470	8.9036	10.6019	37.3522	20.9828	34.5930
	0.3	4.3534	8.5460	10.2314	35.6298	20.0289	28.5904
	0.4	4.1317	8.1890	9.7719	32.8405	18.5098	25.4502
	0.5	3.8501	7.6828	9.1469	28.5379	16.8207	22.1215

	0.6	3.4658	6.9167	8.2511	22.6150	14.6360	18.3096
	0.7	2.9227	5.6916	7.0393	15.7494	11.3399	13.5898
	0.8	2.0447	3.9916	5.2446	8.5825	6.7671	8.2751
	0.9	0.7964	1.8524	2.4622	2.5271	2.2770	2.7962
30	0.1	4.5748	7.5836	8.6234	55.8829	33.5297	16.9680
	0.2	4.4401	7.3500	8.3754	49.5072	27.7576	17.3627
	0.3	4.3107	7.1037	8.1304	43.9779	23.5592	18.1673
	0.4	4.1544	6.8147	7.8632	38.6838	20.385	18.1070
	0.5	3.9541	6.4499	7.5415	32.5252	17.6698	16.9912
	0.6	3.6682	5.9232	7.0117	25.3518	15.0343	14.6416
	0.7	3.2200	5.0920	6.0973	17.4078	12.0248	11.1572
	0.8	2.5077	3.7320	4.5605	9.5027	7.9056	6.9127
	0.9	1.1902	1.7558	2.0321	2.8323	2.9166	2.5581

Table 5 has the same structure as of Table 3. The gap values are calculated using the expression (15). Each class is a set of 20 instances, based on the number of jobs (n), setup range (R), and level of learning factor (b), and then the average of each class is given. The gaps are shown in columns R_i corresponding to both models and that are averaged over each class. From these results we can conclude that the learning effect affects in a different way to each manufacturing environment. While it is true that, in both environments, the gap is inversely proportional to b they are much larger for Model IV. This can be better observed when $n = 25$ and 30.

A graphical representation of the gaps behaviour (on average) for each level of b and each setup range R for the *TCT* objective function is presented in Figure 2. Then, the gaps are calculated by the expression (15) and have been grouped into subsets of 80 values according to R and b.

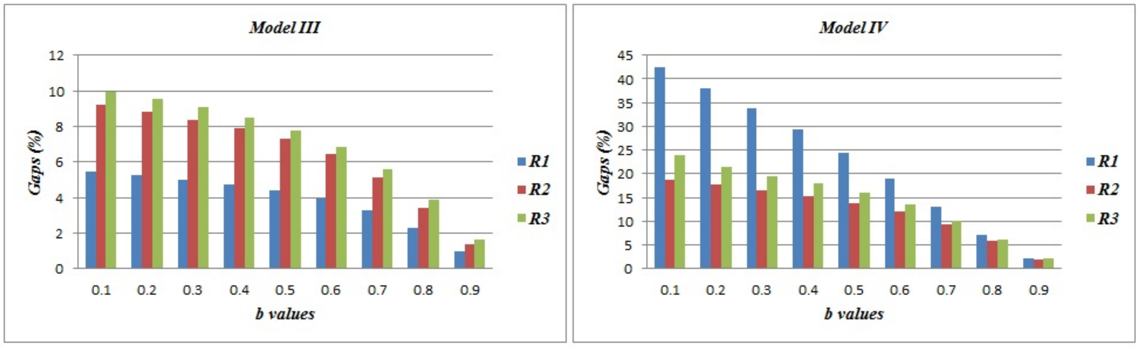

FIGURE 2. Average gaps for each setup time range and each b value.

For Model III (learning effect only on the setup times), the gaps are proportional to the setup range, for each level of b. In other words, the solution without learning effect is degraded when the range of variation of the setup times is greater than the range of variation of the processing times (R_3) for all levels of the learning factor. However, for Model IV (learning effect on both the setup times and the processing times), the quality of the solutions without learning effect is lower when the range of variation of the processing times is greater than the range of variation of the setup times (R_1). It can be observed that the lower values of the gaps are obtained when the processing times and setup times have the same range of variation (R_2). In addition, in both manufacturing environments, the gaps is inversely proportional to b values, i.e., the gaps grow with the learning rate.

Thus, from the experimental study, on the one hand, we can conclude that for the makespan the job position effect on setup times could be ignored and the effect on the quality of the solution would be minute, while for the *TCT* it is vital to consider the job position effect on setup times in the production programming process. On the other hand, when the job position effect affects both setup times and processing times, the solutions obtained without taking into account this effect, for both the C_{max} and the *TCT*, have an inferior solution. The effects are more pronounced for the makespan. In the manufacturing environment our study suggests, for C_{max} and *TCT*, that the job positions should be considered in the production programming process. Additionally, it was observed that, the solutions worsen as the range of setup times decreases.

The results of this study allow decision makers to decide whether or not to include the position-dependent learning effect in the production planning to make more efficient the production process. If inclusion is necessary, the proposed models can be used to obtain prime solutions when the number of batches to be scheduled is 30 or less.

5. Conclusions

Single machine scheduling problem with sequence-dependent setup times is studied in this article. The setup times and the processing times are affected by the job positions in the processing order. We studied the cases of the position-dependent learning effect affects only the setup times, and affects both setup times and processing times. Two performance measures, the makespan and the total completion time, are employed. To tackle the problem, four mathematical formulations are proposed and validated the efficiency of the models over a set of instances adapted from the literature.

In future work, we will extend design of a heuristic algorithms to tackle for larger instances considering position-dependent learning effect in the production programming process. The generalisation of this study to other scheduling problems with sequence-dependent setup times is also another realm of research.

Acknowledgement

We gratefully acknowledge the partial support of the Spanish Ministry of Economy and Competitiveness (Project ECO 2016-76567-C4-2-R).

Conflict of Interests

The author(s) declare that there is no conflict of interests.

References

[1] A. Allahverdi, The third comprehensive survey on scheduling problems with setup times/costs, Eur. J. Oper. Res. 246 (2015), 345–378.

[2] A. Allahverdi and H. Soroush, The significance of reducing setup times/setup costs, Eur. J. Oper. Res. 187 (2008), 978 – 984.

[3] F. Angel-Bello, A. Alvarez, and I. Garc´i a, Two improved formulations for the minimum latency problem, Appl. Math. Model. 37 (2013), 2257–2266.

[4] O. Avalos-Rosales, F. Angel-Bello, A. Alvarez, and Y. Cardona-Valdes, Including preventive maintenance activities in an unrelated parallel machine environment with dependent setup times, Comput. Ind. Eng. 123 (2018), 364 – 377.

[5] A. Azzouz, M. Ennigrou, and L. B. Said, Scheduling problems under learning effects: classification and cartography, Int. J. Product. Res. 56 (2018), 1642–1661.

[6] D. Biskup, Single-machine scheduling with learning considerations, Eur. J. Oper. Res. 115 (1999), 173 – 178.

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

[8] M. Cheng, P. R. Tadikamalla, J. Shang, and B. Zhang, Single machine scheduling problems with exponentially time-dependent learning effects, J. Manuf. Syst. 34 (2015), 60 – 65.

[9] Z. Friggstad, M. Salavatipour, and Z. Svitkina, Asymmetric traveling salesman path and directed latency problems, SIAM J. Comput. 42 (2013), 1596–1619.

[10] S. Gawiejnowicz, Time-dependent scheduling, Springer, Berlin, 2008.

[11] Z. Jiang, F. Chen, and X. Zhang, Single-machine scheduling with times-based and job-dependent learning effect, J. Oper. Res. Soc. 68 (2017), 809–815.

[12] C. Koulamas and G. J. Kyparisis, Single-machine scheduling problems with past-sequence-dependent setup times, Eur. J. Oper. Res. 187 (2008), 1045–1049.

[13] J.-Y. Kung and M.-H. Shu, Some scheduling problems on a single machine with general job effects of position-dependent learning and start-time-dependent deterioration, Asia-Pac. J. Oper. Res. 32 (2015), p. 1550002.

[14] W.-H. Kuo and D.-L. Yang, Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect, Eur. J. Oper. Res. 174 (2006), 1184 – 1190.

[15] L. Li and X.-Y. Wang, Research on single-machine scheduling with a time-dependent learning effect, Int. J. Product. Econ. 170 (2015), 370 – 371.

[16] S. Mustu and T. Eren, The single machine scheduling problem with sequence-dependent setup times and a learning effect on processing times, Appl. Soft Comput. 71 (2018), 291 – 306.

[17] R. Roberti and P. Toth, Models and algorithms for the asymmetric traveling salesman problem: an experimental comparison, Euro J. Transport. Log. 1 (2012), 113–133.

[18] R. Rudek, Computational complexity of the single processor makespan minimization problem with release dates and job-dependent learning, J. Oper. Res. Soc. 65 (2014), 1170–1176.

[19] K. Rustogi and V. A. Strusevich, Simple matching vs linear assignment in scheduling models with positional effects: A critical review, Eur. J. Oper. Res. 222 (2012), 393 – 407.

[20] K. Rustogi and V. A. Strusevich, Single machine scheduling with a generalized job-dependent cumulative effect, J. Sched. 20 (2017), 583–592.

[21] H. M. Soroush, On the scheduling with past-sequence-dependent setup times and learning effects on a single machine, Int. J. Adv. Manuf. Technol. 68 (2013), 2483–2487.

[22] H. M. Soroush, Scheduling in bicriteria single machine systems with past-sequence-dependent setup times and learning effects, J. Oper. Res. Soc. 65 (2014), 1017–1036.

[23] M. D. Toksari and O. A. Arik, Single machine scheduling problems under position-dependent fuzzy learning effect with fuzzy processing times, J. Manuf. Syst. 45 (2017), 159 – 179.

[24] J.-B. Wang and J.-X. Li, Single machine past-sequence-dependent setup times scheduling with general position-dependent and time-dependent learning effects, Appl. Math. Model. 35 (2011), 1388 – 1395.