SOME SCHEDULING PROBLEMS WITH SUM OF LOGARITHM PROCESSING TIMES BASED LEARNING EFFECT AND EXPONENTIAL PAST SEQUENCE DEPENDENT DELIVERY TIMES

MEHMET DURAN TOLSAI∗, EMEL KIZILKAYA AYDOGAN, BERRIN ATALAY AND SAZİYE SARI
Department of Industrial Engineering, Faculty of Engineering, Erciyes University, 38039, Turkey

(Communicated by Gerhard-Wilhelm Weber)

Abstract. In recent years, significant on the past sequence dependent delivery times have been increasing for scheduling problems. An electronic component when waiting to process may be exposed to certain an electromagnetic field and is required to neutralize the effect of electromagnetism. In this case, it needs an extra time to eliminate adverse effect. In the scheduling literature, this extra time is called as past-sequence-dependent delivery times. In this paper we introduce single-machine scheduling problems with an exponential sum-of-actual-processing-time-based delivery times. By the exponential sum-of-actual-processing-time-based delivery times, we mean that the delivery times are defined by an exponential function of the sum of the actual processing times of the already processed jobs. On the other hand, the learning effect is reflected in decreasing processing times based on the job’s position in schedule. In this paper, we also introduce both exponential past sequence dependent delivery times and learning effect where the job processing time is a function based on the sum of the logarithm of processing times of jobs already processed. We show that the single-machine scheduling problems to minimize makespan, total completion time, weighted total completion time and maximum tardiness with sum of logarithm processing times based learning effect and exponential past sequence dependent delivery times have polynomial time solutions.

1. Introduction. Many researchers assume what the processing time is constant. However it may be under the some influences due to various real life applications and/or many practical settings. In this case, machines/workers can develop their performance by repeating similar operations, and the actual processing time of a job is shorter if it is scheduled later in a sequence. This phenomenon is known as the “learning effect” in the literature [1]. In this paper, we use sum of logarithm processing times based learning effect where the job’s actual processing time is a function of the sum of the logarithm of the processing times of the jobs already processed.

2020 Mathematics Subject Classification. Primary: 58F15, 58F17; Secondary: 53C35.

Key words and phrases. Scheduling, learning effect, exponential past sequence dependent delivery times, logarithm processing times, single machine.

The first author is supported by Scientific Research Fund of Erciyes University grant: FBA-2014-5397.

∗ Corresponding author: Mehmet Duran Toksari.
On the other hand, scheduling research has increasingly considered past sequence dependent delivery times. An extra time in the scheduling under the some influences should require. This extra time is modeled as past sequence dependent delivery times and is performed immediately after component has been processed on the machine to provide a delivery to customer [7]. In the scheduling problems, Koulamas and Kyparisis [7] introduce past sequence dependent delivery times. It is proportional to the job’s waiting time for analytical suitability [10].

Cheng et al. [2] presents sum-of-logarithm processing-times-based learning effect for some single machine problems. Xingong et al. [21] studied scheduling problems with two different learning effects such as sum-of-logarithm-processing-time-based learning effect and position-based learning effect. Cheng et al. [3] used position-weighted learning effect based on both sum-of-logarithm-processing-times and job position for processing time of a job. Yin et al. [19] presents some scheduling problems under deterioration based on sum-of-logarithm-processing-times. Wang et al. [16] worked exponential form of the sum-of-logarithm-processing-times based learning effect for some single machine scheduling problems.

Liu et al. [11] studied on preemptive and non-preemptive models of the total completion time on a single machine under both past sequence dependent delivery times and release times. Liu et al. [12] proposed polynomial algorithms for some scheduling problems under deterioration effect and past sequence dependent delivery times. Liu [13] used polynomial algorithms to minimize optimally some identical parallel machine scheduling problems. Shen and Wu [15] investigated some scheduling problems with past sequence dependent delivery times under general position dependent and time dependent leaning effects. This topic has been studied by many people in the field of scheduling. Miao and Zou [4] deals with the problem of single-machine scheduling with simple deterioration and past sequence delivery times. In their study they show that minimization of makespan, minimization of total completion time and minimization of total lateness are solvable polynomial time. They also presented two polynomial time algorithms for parallel-batch scheduling problem.

Wu et al. [20] considered past sequence dependent delivery times and truncated sum-of-processing-times learning effect simultaneously for single machine problem. They proved that their proposed model solves in polynomial time for minimization in some single machine scheduling problems. Li [14] considered nonlinear setup time and delivery times together in his study. Presented by Li optimum scheduling rules for makespan, total completion times, total weighted completion times, sum of the δ-th power of job completion times and maximum tardiness. Wang [17] investigated the single machine scheduling problem related to time-dependent learning effect and setup times. It showed that problems can be solved in polynomial time by using the objective functions of total completion time minimization, weighted total completion time minimization and minimization of maximum tardiness.

Wang et al. [18] studied for single machine scheduling problems, and show that the sum of the makepan minimization problem, the total completion time minimization problem, and the minimization problem of the quadratic job completion times can be solved with SPT rule. They also show that the total weighted completion time minimization problem and the maximum tardiness minimization problem can be solved in polynomial time under certain conditions. Lee [9] considered the learning effect and setup time, a scheduling model has been considered. It has been shown that the proposed model for two single machines can be solved in
polynomial time for problems with objective function processing time, total completion time minimization. It has also shown that it can be solved polynomially for weighted total completion time, maximum tardiness and total tardiness problems. Kuo and Yang [8] considered the setup time and learning effect for the single machine scheduling problem. The objective functions they used were total completion time minimization, absolute difference in completion times and different objective functions. Hsu et. al [6] showed that the algorithm they proposed provides solutions for these problems in polynomial time. worked on parallel machine scheduling problems under the of setup time and learning effect. their objection function is the minimization of total completion time. They showed that the proposed algorithm solves the problem in polynomial time.

In this paper, we introduce both exponential past sequence dependent delivery times and sum of the logarithm of processing times based learning effect.

The rest of this paper is organized as follows. Section 2 presents problem description. In Section 3, we consider several single machine problems under exponential past sequence dependent delivery times and sum of the logarithm of processing times based learning effect. The last section summarizes the conclusions of the research.

2. Problem description. We assume that a set of n jobs is available for processing at time zero on a single machine and they are independent and non-preemptive. \( p_{j[r]} \) denotes the actual processing time of job \( J \) scheduled in position \( r \), \( p_j \) is basic processing time, \( a (a < 0) \) is the learning rate. We consider the following model for actual processing time:

\[
p_{j[r]} = p_j \left( 1 + \sum_{l=1}^{r-1} \ln p_{[l]} \right)^a \quad (j = 1, \ldots, n)
\]  

(1)

On the other hand, the processing of job \( p_{j[r]} \) must be follow the past sequence dependent delivery time \( q_{j[r]} \), which can be formulated as:

\[
q_{j[r]} = \gamma \left( 1 + \sum_{l=1}^{r-1} \ln p_{[l]} \right)^b \quad (j = 1, \ldots, n)
\]  

(2)

where \( \gamma (0 \leq \gamma \leq 1) \) is a normalizing constant, and \( b (b > 1) \) is the index of delivery times. This paper presents the minimization of follow scheduling objectives: makespan \( (C_{max} = \max \{ C_j | j = 1, 2, \ldots, n \}) \), total completion time \( (\sum C) \), weighted total completion time \( (\sum wC) \) and maximum tardiness \( (T_{max}) \) with sum of logarithm processing times based learning effect and exponential past sequence dependent delivery times. We denote all problems using three field notation scheme \( a | \beta | \gamma \) where \( a \) is machine environment, \( \beta \) is job characteristic and \( \gamma \) is optimality criteria [5].

3. Single machine scheduling problems. Let \( (C_{max} = \max \{ C_j | j = 1, 2, \ldots, n \}) \), \( (\sum C) \), \( (\sum wC) \) and \( (T_{max}) \) represent makespan, total completion time, weighted total completion time and maximum tardiness of a given permutation, respectively.

First, we give two lemmas; they are useful for the following theorems.

Lemma 3.1.

\[
(1 - \lambda) + \lambda (1 + \delta x)^a - (1 + \delta \ln \lambda + \delta x)^a + \frac{\gamma \delta^a \left( (\sigma + e^x)^b - (\sigma + \lambda e^x)^b \right)}{e^x} \leq 0
\]  

(3)
Lemma 3.2.  

\[ a < 0, b > 1, 0 < \gamma < 1, \lambda \geq 1, 0 < \delta \leq 1 \quad \text{and} \quad x \geq 1. \]

Proof.  

\[ h(\lambda) = (1 - \lambda) + \lambda (1 + \delta x)^a - (1 + \delta ln\lambda + \delta x)^a + \frac{\gamma \delta a (\sigma + e^x)^b - (\sigma + \lambda e^x)^b}{e^x} \]  

(4)  

Taking the first and the second derivative of \( h(\lambda) \) with respect to \( \lambda \), we have  

\[ h'(\lambda) = -1 + (1 + \delta x)^a - \frac{a \delta (1 + \delta ln\lambda + \delta x)^{a-1}}{\lambda} - \frac{\gamma \delta b e^x (\sigma + \lambda e^x)^{b-1}}{e^x} \]  

(5)  

and  

\[ h''(\lambda) = \frac{a \delta^2 (a - 1) (1 + \delta ln\lambda + \delta x)^{a-2} - a \delta (1 + \delta ln\lambda + \delta x)^{a-1}}{\lambda^2} - \frac{\gamma \delta b (b - 1) e^x (\sigma + \lambda e^x)^{b-2}}{e^x} \]  

(6)  

Since \( a < 0, b > 1, 0 < \gamma < 1, \lambda \geq 1, 0 < \delta \leq 1 \) and \( x \geq 1 \), it implies that \( h''(\lambda) \leq 0 \). Using the fact that \( h'(\lambda) \) is a decreasing function for \( \lambda \geq 1 \). This implies that \( h'(\lambda) \leq h'(1) \leq 0 \). This completes the proof. \( \square \)

**Lemma 3.2.**  

\[ (1 - \lambda) e^x + \lambda e^x (B + x)^a - \lambda_1 e^x (B + \lambda x + x)^a + \lambda_2 \gamma \delta a (\sigma + e^x)^b - \lambda_1 \gamma \delta a (\sigma + \lambda e^x)^b \leq 0 \]  

(7)  

\[ a < 0, b > 1, B > 0, 0 < \gamma < 1, \lambda \geq 1, 0 < \lambda_1, \lambda_2 < 1, 0 < \delta \leq 1 \quad \text{and} \quad x \geq 1. \]

Proof.  

\[ h(\lambda) = (1 - \lambda) e^x + \lambda e^x (B + x)^a - \lambda_1 e^x (B + \lambda x + x)^a + \lambda_2 \gamma \delta a (\sigma + e^x)^b - \lambda_1 \gamma \delta a (\sigma + \lambda e^x)^b \]  

(8)  

Taking the first and the second derivative of \( h(\lambda) \) with respect to \( \lambda \), we have  

\[ h'(\lambda) = -e^x + \lambda_2 e^x (B + x)^a - \frac{\lambda_1 e^x (B + \lambda x + x)^a}{\lambda} - \frac{\lambda_1 \gamma e^x \delta b (\sigma + \lambda e^x)^{b-1}}{\lambda^2} \]  

(9)  

and  

\[ h''(\lambda) = -\frac{\lambda_1 e^x \delta a a (a - 1) (B + \lambda x + x)^{a-2} - \lambda_1 e^x a (B + \lambda x + x)^{a-1}}{\lambda^2} - \frac{\lambda_1 \gamma e^x \delta b (b - 1) (\sigma + \lambda e^x)^{b-2}}{\lambda^2} \]  

(10)  

Since \( a < 0, b > 1, 0 < \gamma < 1, \lambda \geq 1, 0 < \lambda_1, \lambda_2 < 1, 0 < \delta \leq 1 \) and \( x \geq 1 \), it implies that \( h''(\lambda) \leq 0 \). Using the fact that \( h'(\lambda) \) is a decreasing function for \( \lambda \geq 1 \). This implies that \( h'(\lambda) \leq h'(1) \leq 0 \). This completes the proof.

**Notations:**  

\( n \): number of jobs  
\( p_{j[r]} \): actual processing time of job \( j \) scheduled in position \( r \)  
\( p_i \): basic processing time of job \( j \)  
\( a \): the learning rate  
\( q_{j[r]} \): past sequence dependent delivery time of job \( j \) scheduled in position \( r \)  
\( b \): index of delivery times  
\( \gamma \): normalizing constant for past sequence dependent delivery time
π: schedule π = [S_1, J_i, J_j, S_2]
π': schedule π' = [S_1, J_i, J_j, S_2]

C_{j[r]}: completion time of job j scheduled in position r
T_{j[r]}: tardiness of job j scheduled in position r

A: sum of the processing times of the jobs scheduled before the J_i and J_j

Theorem 3.3. For the problem 1/p_{j[r]} = \left(1 + \sum_{l=1}^{r-1} \ln p_l\right)^a,
q = \gamma \left(1 + \sum_{l=1}^{r-1} p_l\right)^b / C_{\text{max}} the optimal schedule can be obtained by sequencing the jobs in non-decreasing order of p_j (the SPT rule).

Proof. Consider an optimal schedule π. Assume under π there are two adjacent jobs, J^i and J^j, such that p^i ≤ p^j and J^j is scheduled directly after J^i in the r-th position in a sequence. In schedule π, we assume that γ ≥ 0 is a normalizing constant, p^j is basic processing time, a(a < 0) is the learning rate and b(b > 1) is the index of delivery times. Let A be the sum of the processing times of the jobs scheduled before the J^i and J^j. One can easily compute C_{i[r]}(π), C_{j[r+1]}(π), C_{j[r]}(π') and C_{i[r+1]}(π') as

\begin{equation}
C_{i[r]}(\pi) = A + p_i \left(1 + \sum_{l=1}^{r-1} \ln p_l\right)^a \tag{11}
\end{equation}

\begin{align}
C_{j[r+1]}(\pi) &= A + p_i \left(1 + \sum_{l=1}^{r-1} \ln p_l\right)^a + p_j \left(1 + \sum_{l=1}^{r-1} \ln p_l\right)^a + \\
&\quad \gamma \left(1 + \sum_{l=1}^{r-1} \ln p_l + p_i\right)^b \tag{12}
\end{align}

By performing a pairwise interchange on jobs J^j and J^i, we obtain schedule π'.

\begin{equation}
C_{j[r]}(\pi') = A + p_j \left(1 + \sum_{l=1}^{r-1} \ln p_l\right)^a \tag{13}
\end{equation}

\begin{align}
C_{i[r+1]}(\pi') &= A + p_j \left(1 + \sum_{l=1}^{r-1} \ln p_l\right)^a + p_i \left(1 + \sum_{l=1}^{r-1} \ln p_l + \ln p_j\right)^a + \\
&\quad \gamma \left(1 + \sum_{l=1}^{r-1} \ln p_l + p_j\right)^b \tag{14}
\end{align}

After taking the difference between Equations (2) and (12), we have
Theorem 3.4. For the problem

the optimal schedule will be in the SPT order. This completes the proof.

Let \( \lambda = p_j/p_i, \ln \lambda = \ln p_j - \ln p_i \) and \( \ln \lambda + x = \ln p_j \)
\( x = \ln p_i \) then \( e^x = p_i, \delta = 1/1 + \sum_{i=1}^{r-1} \ln p_i \) then \( \delta = 1/B \) and \( \sigma = 1 + \sum_{i=1}^{r-1} \ln p_i \)
we have

\[
\frac{C_{\max}(\pi) - C_{\max}(\pi')}{p_i(B)^a} = (1 - \lambda) + \lambda (1 + \delta x)^a - (1 + \delta \ln \lambda + \delta x)^a + \\
\gamma \delta^a (\sigma + e^x)^b - (\sigma + \lambda e^x)^b
\]

Since \( a < 0, b > 1, B > 0, 0 < \gamma < 1, \lambda \geq 1, 0 < \delta \leq 1 \) by Lemma 3.1, we have
\[
\frac{C_{\max}(\pi) - C_{\max}(\pi')}{p_i(B)^a} \leq 0
\]
So we obtain \( C_{\max}(\pi) \leq C_{\max}(\pi') \) which contradicts the optimality of \( \pi \). Hence, the optimal schedule will be in the SPT order. This completes the proof.

\[\Box\]

Theorem 3.4. For the problem \( 1/p_j[r] = \left(1 + \sum_{i=1}^{r-1} \ln p_i \right)^a \),
\[ q = \gamma \left(1 + \sum_{i=1}^{r-1} \ln p_i + p_j \right)^b \]
the optimal schedule can be obtained by sequencing the jobs in non-decreasing order of \( p_j \) (the SPT rule).

Proof. Here, we still use the same notations as in the proof of Theorem 3.3. The proof of Theorem 3.4 is similar to the proof of Theorem 1 and we know \( C_i[r] \), \( C_j[r+1] \), \( C_j[r] \) and \( C_i[r+1] \) from Theorem 3.5. then Based on Eq. (12) and Eq. (13), we find

\[
\sum C(\pi) - \sum C(\pi') = 2 \left(p_i - p_j\right) \left(1 + \sum_{i=1}^{r-1} \ln p_i \right)^a + p_j \left(1 + \sum_{i=1}^{r-1} \ln p_i + p_i \right)^a - \\
p_i \left(1 + \sum_{i=1}^{r-1} \ln p_i + p_j \right)^a + \gamma \left(1 + \sum_{i=1}^{r-1} \ln p_i + p_i \right)^b - \\
\gamma \left(1 + \sum_{i=1}^{r-1} \ln p_i + p_j \right)^b
\]

Let \( \lambda = p_j/p_i, \ln \lambda = \ln p_j - \ln p_i \) and \( \ln \lambda + x = p_j \)
\( x = \ln p_i \) then \( e^x = p_i, \delta = 1/1 + \sum_{i=1}^{r-1} \ln p_i \) then \( \delta = 1/B \) and \( \sigma = 1 + \sum_{i=1}^{r-1} \ln p_i \)
we have

\[
\frac{C_{\max}(\pi) - C_{\max}(\pi')}{p_i(B)^a} = 2 \left(1 - \lambda\right) + \lambda \left(1 + \delta x\right)^a - \left(1 + \delta \ln \lambda + \delta x\right)^a +
\]

\[\Box\]
Theorem 3.5. For the problem \( 1/p_j[r] = \left(1 + \sum_{l=1}^{r-1} \ln p_l\right)^a \),
\[ q = \gamma \left(1 + \sum_{l=1}^{r-1} \ln p_l\right)^b /T_{\text{max}} \]
the optimal schedule can be obtained by sequencing the jobs in non-decreasing order of \( d_j \) (EDD rule).

Proof. In this proof, we still use the same notations as in the proof of Theorem 3.3. Suppose that \( \pi \) and \( \pi' \) are two job schedules. The difference between \( \pi \) and \( \pi' \) is a pairwise interchange of two adjacent jobs \( J_i \) and \( J_j \), i.e., \( \pi = [S_1, J_i, J_j, S_2] \) and \( \pi' = [S_1, J_j, J_i, S_2] \), where \( S_1 \) and \( S_2 \) each denote a partial sequence.

Let \( T_i[r](\pi) \) and \( T_j[r+1](\pi) \) be the objective function values of jobs \( J_i \) and \( J_j \), respectively.

\begin{align*}
T_i[r](\pi) &= A + p_i \left(1 + \sum_{l=1}^{r-1} \ln p_l\right)^a - d_i, 0 \\
T_j[r+1](\pi) &= A + p_i \left(1 + \sum_{l=1}^{r-1} \ln p_l\right)^a + p_j \left(1 + \sum_{l=1}^{r-1} \ln p_l + \ln p_j\right)^a + \\
&\quad \gamma \left(1 + \sum_{l=1}^{r-1} p_l + p_j\right)^b - d_j, 0
\end{align*}

and

\begin{align*}
T_i[r+1](\pi') &= A + p_j \left(1 + \sum_{l=1}^{r-1} \ln p_l\right)^a - d_i, 0 \\
T_j[r+1](\pi') &= A + p_j \left(1 + \sum_{l=1}^{r-1} \ln p_l\right)^a + p_i \left(1 + \sum_{l=1}^{r-1} \ln p_l + \ln p_i\right)^a + \\
&\quad \gamma \left(1 + \sum_{l=1}^{r-1} p_i + p_j\right)^b - d_i, 0
\end{align*}

If \( d_i \geq d_j \) we have \( T_j[r+1](\pi') > T_i[r](\pi) \). In addition, if \( p_i \leq p_j \) and \( d_i \leq d_j \), from Theorem 3.3., we have \( T_j[r+1](\pi) > T_i[r+1](\pi') \). Therefore, we have \( \max\{T_i[r](\pi), T_j[r+1](\pi)\} > \{T_j[r](\pi'), T_i[r+1](\pi')\} \). Hence, the optimal schedule will be in the EDD order. This completes the proof.

Example 1. Consider a set of five jobs where \( p_1 = 8, p_2 = 6, p_3 = 9, p_4 = 12, p_5 = 10, d_1 = 13, d_2 = 15, d_3 = 10, d_4 = 7, d_5 = 9 \) the learning effect is \( a = -0.5 \), \( b \) is the index of delivery times \( b = 1.1 \) and the normalizing constant for delivery \( \gamma = 0.3 \).

According Theorems 3.3, we know that the optimal schedule is \{\( J_2, J_3, J_3, J_5, J_4 \)\} for following objective functions: \( C_{\text{max}}, \sum C \) and \( T_{\text{max}} \).

The actual processing times \( (p_j[r]) \), past sequence dependent delivery times \( (q_{psd}) \), completion times \( (C_j[r]) \) and tardiness for each job \( (T_j[r]) \) are found as Table 1 when we assume that \( C_0 = 0 \).
Table 1. The results for Example 1

| $p_{j[r]}$ | $q_{psd}$ | $C_{i[r]}$ | $T_{j[r]}$ |
|-----------|-----------|------------|------------|
| 6.00      |           | 6          | 0          |
| 4.79      |           | 10.79      | 0          |
| 4.08      |           | 14.87      | 4.87       |
| 3.76      |           | 18.63      | 9.63       |
| 3.92      | 14.51     | 22.5 + 14.51 = 37.06 | 30.06       |

We find the objective functions as $C_{max}(SPT) = 37.06$, $\sum C_j(SPT) = 87.35$ and $T_{max}(EED) = 30.06$. Figure 1 shows all results.

Figure 1. The gantt chart for Example 1

**Theorem 3.6.** For the problem $1/p_{j[r]} = \left(1 + \sum_{l=1}^{r-1} \ln p_{i[l]}\right)^a$, $q = \gamma \left(1 + \sum_{l=1}^{r-1} \ln p_{i[l]}\right)^b / \sum wC$ the optimal schedule can be obtained by sequencing the jobs in non-decreasing order of $p_j/w_j$ (the WSPT rule).

**Proof.** Here, we still use the same notations as in the proof of Theorem 1. The proof of Theorem 3.6 is similar to the proof of Theorem 3.3 and one can easily compute $w_iC_{i[r]}(\pi)$, $w_jC_{j[r]}(\pi)$, $w_jC_{j[r]}(\pi')$ and $w_iC_{i[r]}(\pi')$ as

$$w_iC_{i[r]}(\pi) = w_i \left(A + p_i \left(1 + \sum_{l=1}^{r-1} \ln p_{i[l]}\right)^a\right)$$

(23)
The optimal schedule will be in the WSPT order. This completes the proof.

By Lemma 3.2,

\[
\sum w_j C_{j[r+1]}(\pi) = w_j \left( A + p_i \left( 1 + \sum_{i=1}^{r-1} \ln p_i \right)^a + p_j \left( 1 + \sum_{i=1}^{r-1} \ln p_i + \ln p_i \right)^a \right) + w_j \left( \gamma \left( 1 + \sum_{i=1}^{r-1} p_i + p_j \right)^b \right)
\]

(24)

By performing a pairwise interchange on jobs \( J_j \) and \( J_i \), we obtain schedule \( \pi' \).

\[
w_j C_{j,r}(\pi') = w_j \left( A + p_j \left( 1 + \sum_{i=1}^{r-1} \ln p_i \right)^a \right)
\]

(25)

\[
w_i C_{i[r+1]}(\pi') = w_i \left( A + p_i \left( 1 + \sum_{i=1}^{r-1} \ln p_i \right)^a + p_i \left( 1 + \sum_{i=1}^{r-1} \ln p_i + \ln p_j \right)^a \right) + w_i \left( \gamma \left( 1 + \sum_{i=1}^{r-1} p_i + p_j \right)^b \right)
\]

(26)

Let \( x = \ln p_i \rightarrow e^x = p_i \), \( \delta = 1/1 + \sum_{i=1}^{r-1} \ln p_i \rightarrow \delta = 1/B \), \( \sigma = 1 + \sum_{i=1}^{r-1} \ln p_i \)

We obtain

\[
\sum wC(\pi) = A (w_i + w_j) + p_i (B)^a (w_i + w_j) + w_j p_j (B + x)^a + w_j (\sigma + e^x)^b
\]

(27)

\[
\sum wC(\pi') = A (w_i + w_j) + p_j (B)^a (w_i + w_j) + w_i p_i (B + x)^a + w_i (\sigma + e^x)^b
\]

(28)

Let \( \lambda_1 = \frac{w_i}{w_i + w_j} \), \( \lambda_2 = \frac{w_j}{w_i + w_j} \), \( \epsilon = w_i + w_j \)

Let \( \lambda = p_j/p_i \), \( \ln \lambda = \ln p_j - \ln p_i \) and \( \ln \lambda + x = p_j \)

After taking the difference between Equations (27) and (28), we have

\[
\frac{\sum wC(\pi) - \sum wC(\pi')}{(w_i + w_j)(B)^a} = (p_i - p_j) + \lambda_2 p_j \delta^a (B + x)^a + \lambda_1 p_i \delta^a (B + \ln \lambda + x)^a
\]

\[
+ \lambda_2 \gamma \delta^a (\sigma + e^x)^b - \lambda_1 \gamma \delta^a (\sigma + \epsilon \sigma)^b
\]

(29)

\[
\frac{\sum wC(\pi) - \sum wC(\pi')}{(w_i + w_j)(B)^a} = (1 - \lambda) e^x + \lambda_2 \lambda \epsilon \delta^a (B + x)^a - \lambda_1 e^x \delta^a (B + \ln \lambda + x)^a
\]

\[
+ \lambda_2 \gamma \delta^a (\sigma + e^x)^b - \lambda_1 \gamma \delta^a (\sigma + \epsilon \sigma)^b
\]

(30)

Since \( a < 0, b > 1, B > 0, 0 < \gamma < 1, \lambda \geq 1, 0 < \lambda_1, \lambda_2 < 1, 0 < \delta \leq 1, x \geq 1 \)

by Lemma 3.2,

we have \( \sum wC(\pi) - \sum wC(\pi') \leq 0 \).

So we obtain \( \sum wC(\pi) \leq \sum wC(\pi') \) which contradicts the optimality of \( \pi \). Hence, the optimal schedule will be in the WSPT order. This completes the proof. \( \square \)
Example 2. Consider a set of five jobs where \( p_1 = 8, p_2 = 6, p_3 = 9, p_4 = 12, p_5 = 10, w_1 = 9, w_2 = 13, w_3 = 7, w_4 = 3, w_5 = 5 \) the learning effect is \( a = -0.5 \), \( b \) is the index of delivery times \( b = 1.1 \) and the normalizing constant for delivery \( \gamma = 0.3 \). According Theorems 3.3, we know that the optimal schedule is \( \{J_2, J_1, J_3, J_5, J_4\} \) for \( \sum w C \). The actual processing times \( (p_j) \), past sequence dependent delivery times \( (p_{psd}) \), completion times \( (C_j) \), and \( w_j(C_j) \) (weighted completion times) are found as Table 2 when we assume that \( C_0 = 0 \).

| \( p_j \) | \( q_{psd} \) | \( C_j \) | \( T_j \) |
|---|---|---|---|
| \( p_{21} \) | 6.00 | \( C_{21} \) | 6 | \( T_{21} \) |
| \( p_{22} \) | 4.79 | \( C_{11} \) | 10.79 | \( T_{11} \) |
| \( p_{33} \) | 4.08 | \( C_{33} \) | 14.87 | \( T_{33} \) |
| \( p_{54} \) | 3.76 | \( C_{54} \) | 18.63 | \( T_{54} \) |
| \( p_{45} \) | 3.92 | \( q_{psd} \) | 14.51 | \( C_{45} \) | \( C_{45} = 22.5 + 14.51 = 37.06 \) | \( T_{45} \) |

We find the objective function as \( \sum w_j C_j(SPT) = 483.53 \). Figure 2 shows all results.

![Gantt Chart](image)

Figure 2. The gantt chart for Example 2

Practical Application (Laptop Parts). An example of laptop parts is given. In this process, some parts may be exposed to certain electromagnetic and/or radioactive fields. Some regulatory authorities may be required while these parts are waiting, and this case will cause the extra time. Processing times and due dates are given in Table 3. The learning effect is \( a = -0.5 \), \( b \) is the index of delivery times \( b = 1.1 \) and the normalizing constant for delivery \( \gamma = 0.3 \). According Theorems 3.3, we know that the optimal schedule is \( \{J_2, J_3, J_4, J_6, J_1, J_7, J_9, J_8, J_5\} \) for following objective functions: \( C_{max} \), \( \sum C \) and \( T_{max} \). The actual processing times \( (p_j) \), past sequence \( (p_{psd}) \) dependent delivery times \( (C_j) \), completion times \( (C_j) \) and tardiness for each job \( (T_j) \) are found as Table 3 when we assume that \( C_0 = 0 \).

We find the objective functions as \( C_{max}(SPT) = 153.57 \), \( \sum C(SPT) = 376.10 \) and \( T_{max}(EDD) = 148.57 \). Figure 3 shows all results.

![Table 3](image)

Table 3. The results for Practical Application (Laptop Parts)
Table 3. Laptop components of processing times and due dates

| Job | Part Name | Processing Time | Due Date |
|-----|-----------|-----------------|----------|
| 1   | CPU/GPU   | 24              | 9        |
| 2   | IC Chip   | 6               | 35       |
| 3   | Oscillator| 9               | 25       |
| 4   | Copper coil| 12          | 20       |
| 5   | Capacitor | 70              | 5        |
| 6   | Card slot | 15              | 10       |
| 7   | Ports     | 25              | 8        |
| 8   | Cooling fan| 42            | 6        |
| 9   | Sub-PCB   | 35              | 7        |

Table 4. The results for Example 2

| \( p_j \) | \( q_{ped} \) | \( C_{i[r]} \) | \( T_{i[r]} \) |
|----------|---------------|---------------|---------------|
| \( p_2[1] \) | 6.00 | \( C_{2[1]} \) | 6 | \( T_{2[1]} \) | 0 |
| \( p_3[2] \) | 5.39 | \( C_{3[2]} \) | 11.39 | \( T_{3[2]} \) | 0 |
| \( p_4[3] \) | 5.37 | \( C_{4[3]} \) | 16.76 | \( T_{4[3]} \) | 0 |
| \( p_6[4] \) | 5.49 | \( C_{6[4]} \) | 22.25 | \( T_{6[4]} \) | 12.25 |
| \( p_1[5] \) | 7.52 | \( C_{1[5]} \) | 29.77 | \( T_{1[5]} \) | 20.77 |
| \( p_7[6] \) | 6.84 | \( C_{7[6]} \) | 36.61 | \( T_{7[6]} \) | 28.61 |
| \( p_9[7] \) | 8.60 | \( C_{9[7]} \) | 45.20 | \( T_{9[7]} \) | 38.20 |
| \( p_{8}[8] \) | 9.36 | \( C_{8[8]} \) | 54.56 | \( T_{8[8]} \) | 48.56 |
| \( p_{9}[9] \) | 14.33 | \( q_{ped} \) | 84.68 | \( C_{9[9]} \) | 68.89+84.68=153.57 | \( T_{9[9]} \) | 63.89 |

Figure 3. The Gantt chart for the practical application

4. Conclusion. In this paper, we present some single machine scheduling problems with exponential past sequence dependent delivery times under learning effect where the job processing time is a function based on the sum of the logarithm of processing times of jobs already processed. The actual processing time of a given job drops to zero precipitously as the number of jobs increases in the position-based
model or when the normal job processing times are large in the sum-of-processing-times-based model. Motivated by this observation, we used the sum of the logarithm of the processing times of the jobs already processed. The objectives consist of minimizing makespan, total completion time, maximum tardiness and weighted total completion time. We prove that minimizing the makespan and total completion time can be performed by sequencing the jobs according to the shortest normal processing time first (SPT) rule. We also show that the agreeable maximum tardiness minimization problem can be solved optimally by sequencing jobs in non-decreasing order of their due dates (EDD rule). Finally, we present that minimizing the weighted total completion time can be performed by sequencing the jobs according to the weighted shortest normal processing time first (WSPT) rule.

In particular, we showed that the problems under study to minimize on a single-machine with exponential past sequence dependent delivery times are polynomially solvable under sum of logarithm processing times based learning effect. The result shows that future researches may focus on single/multi machine scheduling problems with exponential past sequence dependent delivery times under the different combinations of learning effect and deterioration effect.

Acknowledgments. This research was supported by Scientific Research Fund of Erciyes University under the contract no: FBA-2014-5397.

REFERENCES

[1] A. B. Badiru, Computational survey of univariate and multivariate learning curve models, IEEE Transactions on Engineering Management, 39 (1992), 176–188.
[2] T. C. E. Cheng, P. Lai, C. Wu and W. Lee, Single-machine scheduling with sum-of-logarithm-processing-times-based learning considerations, Information Sciences, 179 (2009), 3127–3135.
[3] T. C. E. Cheng, W. Kuo and D. Yang, Scheduling with a position-weighted learning effect based on sum-of-logarithm-processing-times and job position, Information Sciences, 221 (2013), 490–500.
[4] C. Miao and J. Zou, Scheduling problem with simple deterioration and past-sequence-dependent delivery times, Operations Research Transactions, 20 (2016), 61–68.
[5] R. L. Graham, E. L. Lawler, J. K. Lenstra and A. H. G.Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: A Survey, Annals of Discrete Mathematics, 5 (1979), 287–326.
[6] C. Hsu, W. Kuo and D. Yang, Unrelated parallel machine scheduling with past-sequence-dependent setup time and learning effects, Applied Mathematical Modelling, 35 (2011), 1492–1496.
[7] C. Koulimas and G. J. Kyparisis, Single-machine scheduling problems with past-sequence-dependent delivery times, International Journal of Production Economics, 126 (2010), 264–266.
[8] W. Kuo and D. Yang, Single machine scheduling with past-sequence-dependent setup times and learning effects, Information Processing Letters, 102 (2007), 22–26.
[9] W. Lee, A note on single-machine scheduling with general learning effect and past-sequence-dependent setup time, Computers and Mathematics with Applications, 62 (2011), 2095–2100.
[10] M. Liu, F. Zheng, C.n Chu and Y. Xu, New results on single-machine scheduling with past-sequence-dependent delivery times, Theoretical Computer Science, 438 (2012), 55–61.
[11] M. Liu, F. Zheng, C. Chu and Y. Xu, Single-machine scheduling with past-sequence-dependent delivery times and release times, Information Processing Letters, 112 (2012), 835–838.
[12] M. Liu, S. Wang and C. Chu, Scheduling deteriorating jobs with past-sequence-dependent delivery times, International Journal of Production Economics, 144 (2013), 418–421.
[13] M. Liu, Parallel-machine scheduling with past-sequence-dependent delivery times and learning effect, Applied Mathematical Modelling, 37 (2013), 9630–9633.
[14] L. Mingze, Single-Machine Scheduling Problems with Non-Linear Past-Sequence-Dependent Setup Times and Delivery Times, A a, 1 (2017), 2.
SCHEDULING PROBLEMS WITH SUM OF LOGARITHM PROCESSING TIMES... 13

[15] L. Shen and Y. Wu, Single machine past-sequence-dependent delivery times scheduling with general position-dependent and time-dependent learning effects, *Applied Mathematical Modelling*, 37 (2013), 5444–5451.

[16] J. Wang, L. Sun and L. Sun, Single machine scheduling with exponential sum-of-logarithm-processing-times based learning effect, *Applied Mathematical Modelling*, 34 (2010), 2813–2819.

[17] J. Wang, Single-machine scheduling with past-sequence-dependent setup times and time-dependent learning effect, *Computers and Industrial Engineering*, 55 (2008), 584–591.

[18] J. Wang, D. Wang, L. Wang, L. Lin, N. Yin and W. Wang, Single machine scheduling with exponential time-dependent learning effect and past-sequence-dependent setup times, *Computers and Mathematics with Applications*, 57 (2009), 9–16.

[19] N. Yin, L. Kang, P. Ji and J. Wang, Single machine scheduling with sum-of-logarithm-processing-times based deterioration, *Information Sciences*, 274 (2014), 303–309.

[20] W. Yu-Bin and W. Jian-Jun, Single-machine scheduling with truncated sum-of-processing-times-based learning effect including proportional delivery times, *Neural Computing and Applications*, 27 (2016), 937–943.

[21] X. Zhang, G. Yan, W. Huang and G. Tang, A note on machine scheduling with sum-of-logarithm-processing-time-based and position-based learning effects, *Information Sciences*, 187 (2012), 298–304.

Received September 2020; 1st revision November 2020; 2nd revision December 2020.

E-mail address: dtoksari@erciyes.edu.tr
E-mail address: ekaydogan@erciyes.edu.tr
E-mail address: berrinucarkus@erciyes.edu.tr
E-mail address: saziyesari.end@gmail.com