RAIRO-Oper. Res. 44 (2010) 107–118
DOI: 10.1051/ro/2010009

RAIRO Operations Research
www.rairo-ro.org

SOME SCHEDULING PROBLEMS WITH PAST SEQUENCE DEPENDENT SETUP TIMES UNDER THE EFFECTS OF NONLINEAR DETERIORATION AND TIME-DEPENDENT LEARNING

M. Duran Toksari[1], Daniel Oron[2] and Ertan Güner[3]

Abstract. This paper studies scheduling problems which include a combination of nonlinear job deterioration and a time-dependent learning effect. We use past sequence dependent (p-s-d) setup times, which is first introduced by Koulamas and Kyparisis [*Eur. J. Oper. Res.* **187** (2008) 1045–1049]. They considered a new form of setup times which depend on all already scheduled jobs from the current batch. Job deterioration and learning co-exist in various real life scheduling settings. By the effects of learning and deterioration, we mean that the processing time of a job is defined by increasing function of its execution start time and a function of the total normal processing time of jobs scheduled prior to it. The following objectives are considered: single machine makespan and sum of completion times (square) and the maximum lateness. For the single-machine case, we derive polynomial-time optimal solutions.

Keywords. Scheduling, single Machine, past sequence dependent (p-s-d) setup times, time-dependent learning effect, deterioration jobs.

Mathematics Subject Classification. 90B35.

Received May 5, 2008. Accepted July 26, 2009.

[1] Erciyes University, Engineering Faculty, Industrial Engineering Department, Kayseri, Turkey; dtoksari@erciyes.edu.tr

[2] Econometrics and Business Statistics, The University of Sydney, NSW 2006, Australia.

[3] Gazi University, Engineering and Architecture Faculty, Industrial Engineering Department, Ankara, Turkey.

Article published by EDP Sciences

1. Introduction

Koulamas and Kyparisis [1] first introduced a scheduling problem with past sequence dependent (p-s-d) setup times. They considered a new form of setup times which depend on all already scheduled jobs from the current batch. They showed that the standard single machine scheduling with p-s-d setup times can be solvable in polynomial time when the objectives are makespan, the total completion time and the total absolute differences in completion times, respectively. Kou and Yang [2] studied single machine scheduling with past sequence dependent setup times and learning effects. They proposed polynomial time algorithms to solve makespan, the total completion time, the total absolute differences in completion times and the sum of earliness, tardiness and common due date penalty.

This paper addresses several single machine scheduling problems with past sequence dependent setup times under the assumption of nonlinear effects of learning and deterioration. The time-dependent learning effect of a job is assumed to be a function of total normal processing time of jobs scheduled prior to the execution of this job. Scheduling problems are the core of many manufacturing systems, and have, thus, become an important area of research in recent decades. In classical scheduling theory, job processing times are considered to be constant and independent of earlier processed jobs. In practice, however, we often encounter setting in which processing times increase or decrease as a function of the past sequence of jobs.

Scheduling problems with deterioration jobs have received increasing attention in recent years. Scheduling problems with time-dependent processing times were initiated independently by Gupta and Gupta [3] and Browne and Yechiali [4]. They proposed models, which depend on the processing time function. Alidae and Womer [5] classified deteriorating jobs models into three different types: linear, piecewise linear and non-linear. In this paper, we focus on the latter type, *i.e.*, non-linear deterioration effect.

Up to date research has mainly focused on linear models, while little attention has been given to the non-linear counterpart. Recently, Voutsinas and Pappis [6] introduced a new type of nonlinear deterioration entitled job value, which assumes exponential deterioration over time. The objective is finding a processing sequence of the jobs in such a way that the total value reduction of jobs is minimized. Cheng *et al.* [7] introduced comprehensive reviews of different models and problems concerning jobs with start-time-dependent processing times. In this paper, we consider the nonlinear deterioration effect proposed by Alideaee and Womer [5].

$$P_i\,(t) = p_i + \alpha t_i^b \tag{1.1}$$

where $\alpha(\alpha > 0)$ and $(b > 0)$ is nonlinear deterioration effect, which is the amount of increase in the processing time of a job per unit delay in its starting time.

The common assumption is that machines and workers do not improve their rate of production over time. However, in many realistic settings, workstations improve continuously as a result of repeating the same or similar activities. Thus, the processing time of a job is shorter if it is scheduled later in the sequence. Mosheiov [8] determined that this phenomenon is known in the literature as a "learning effect". The learning effect has been studied in the context of scheduling problems by many researchers in recent years [8–12]. Biskup [9] was the first to investigate the learning effect in the scheduling problems. He assumed that production time of a single item under learning effect decreases as a function of the task's position in the sequence. Kuo and Yang [13] assumed the time-dependent learning effect of a job to be a function of the total normal processing time of jobs scheduled in front of it (see Eq. (1.2)). We use this learning effect in our model.

$$p_{ir} = \left(1 + p_{[1]} + p_{[2]} + \ldots + p_{[r-1]}\right)^a p_i = \left(1 + \sum_{k=1}^{r-1} p_{[k]}\right)^a p_i \tag{1.2}$$

where p_{ir} is actual processing time of the job performed at position r when $p_{[i]}$ is its basic processing time and $a \leq 0$ is learning index.

In the literature, there are a few studies on scheduling problems with effects of learning and deterioration simultaneously. Wang and Cheng [14] studied a single-machine scheduling problem with deteriorating jobs and learning effects to minimize the makespan. Wang [15] developed a polynomial time solution for the single machine scheduling problems with deteriorating jobs and learning effects. Toksari and Guner [16] proposed mixed nonlinear integer model for parallel machine earliness/tardiness scheduling problem with sequence dependent setup time and the effects of learning and deterioration. Cheng *et al.* [17] derived polynomial-time optimal solutions for several scheduling problems with deteriorating jobs and learning effect. Linear deterioration effect was considered in all studies on scheduling problems effects of learning and deterioration simultaneously. In this paper, we address several scheduling problems with nonlinear effect of learning and deterioration simultaneously.

The rest of the paper is organized as follows: in Section 2, we will formulate the model under study. In Section 3, we derive polynomial-time optimal solutions for some single machine scheduling problems with p-s-d setup times under learning effect and nonlinear deteriorating jobs. The conclusions are summarized in Section 4.

2. Problem formulation

We consider that the time-dependent learning effect of a job, which is assumed to be a function of total normal processing time of jobs scheduled in front of it, proposed by Kuo and Yang [13]. It was introduced by Alidaee and Womer [5] to model the effect of job deterioration. In this study, effects of deterioration and learning are considered simultaneously, and above two effects are combined as follows:

$$\hat{p}_r = \left[p_r + \left(\alpha \times t_r^b\right)\right] \left(1 + \sum_{k=1}^{r-1} p_k\right)^a . \tag{2.1}$$

There are n jobs to be scheduled on single machine. If job i, $i = 1, 2, \ldots, n$, is scheduling in position r in a sequence, its actual processing time is $\hat{p}_r$. p_r is basic processing time of job scheduled in position r. $\alpha(\alpha > 0)$ and $(b > 0)$ is nonlinear deterioration effect, which is the amount of increase in the processing time of a job per unit delay in its starting time. $a(a < 0)$ is the learning index. t_r is starting time of job scheduling in position r, and C_{r-1} is the completion time of the job scheduled in position $(r-1)$. Thus, the actual processing time $\hat{p}_r$ is formulized follow;

$$\hat{p}_r = \left[p_r + \left(\alpha\ \times\ C_{r-1}^b\right)\right]\left(1 + \sum_{k=1}^{r-1} p_k\right)^a \tag{2.2}$$

where C_{r-1} is the actual completion time of the job scheduled in position $(r-1)$.

Furthermore, as in Koulamas and Kyparisis [2], it is assumed that setup time (s_r^{psd}) of the job scheduled in position $r(J_r)$ when scheduled in position r is given as follows:

$$s_1^{psd} = 0 \tag{2.3}$$

$$s_r^{psd} = \gamma \sum_{i=1}^{r-1} p_i \tag{2.4}$$

where $\gamma \geq 0$ is a normalizing constant and $\hat{p}_i$ is actual processing time of job performed at position i. The value of the normalizing constant γ determines the actual lengths of the required setups and when $\gamma = 0$ there is no need for any p-s-d setups [2].

3. Some single machine scheduling problems

Let $C_{max} = \max\{C_j\,|j = 1, 2, \ldots, n\}$, $\sum C_j$, $\sum C_j^2$ and $L_{max} = \max\{C_j - d_j|j = 1, 2, \ldots, n\}$ represent the makespan, the sum of completion times, the sum of completion time square and the maximum lateness of a given permutation, respectively.

Theorem 3.1. *The problem* $1\left|\left[\left[p_r + \left(\alpha\ \times\ t_r^b\right)\right]\left(1 + \sum\limits_{k=1}^{r-1} p_k\right)^a\right] + s_r^{\text{psd}}\right|C_{max}$ *can be solved optimally by sequencing jobs in non-decreasing order of their processing times (SPT rule).*

Proof. Consider an optimal schedule π, which contains two adjacent jobs, job J_u followed by job $J_v(v = u+1)$, such that $p_u < p_v$. The starting time of J_u is T and C_u and C_v express completion time of the jobs scheduled in the position $u(J_u)$ and v and (J_v), scheduled at position u and $(v = u+1)$, respectively. With the nonlinear effects of learning and deterioration we obtain:

$$
\begin{aligned}
&C_v(\pi) = T + \left[\left(\left(p_u + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right) + \gamma \sum_{k=1}^{r-1} p_k\right] + \\
&\left[\begin{array}{l}
\left(\begin{array}{l}
\left(p_v + \left(\alpha \times \left[T + \left(\left(p_u + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right) + \gamma \sum_{k=1}^{r-1} p_k\right]^b\right)\right) \times \\
\left(1 + \sum_{k=1}^{r-1} p_k + \left(\left(p_u + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right)\right)^a
\end{array}\right) + \\
\gamma \sum_{k=1}^{r-1} p_k + \gamma\left(\left(p_u + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right)
\end{array}\right].
\end{aligned}
$$

By performing a pairwise interchange on jobs J_u and J_v, we obtain schedule π' where the starting time of J_v is T. The completion times of the jobs processed before jobs J_u and J_v are not affected by interchange, and thus,

$$
\begin{aligned}
&C_u(\pi') = T + \left[\left(\left(p_v + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right) + \gamma \sum_{k=1}^{r-1} p_k\right] + \\
&\left[\begin{array}{l}
\left(\begin{array}{l}
\left(p_u + \left(\alpha \times \left[T + \left(\left(p_v + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right) + \gamma \sum_{k=1}^{r-1} p_k\right]^b\right)\right) \times \\
\left(1 + \sum_{k=1}^{r-1} p_k + \left(\left(p_v + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right)\right)^a
\end{array}\right) + \\
\gamma \sum_{k=1}^{r-1} p_k + \gamma\left(\left(p_v + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right)
\end{array}\right].
\end{aligned}
$$

We substitute $x = \sum_{k=1}^{r-1} p_k$, and, therefore, the difference between the values of $C_v(\pi)$ and $C_u(\pi')$ is:

$$
\begin{aligned}
&C_u(\pi') - C_v(\pi) = \left((1+x)^a \times (p_v - p_u)\right) + \left(p_u \times \left(1 + x + \left(p_v \times (1+x)^a\right) + \right.\right. \\
&\left.\left.\left(\alpha \times T^b (1+x)^a\right)\right)^a\right) - \left(p_v \times \left(1 + x + \left(p_u \times (1+x)^a\right) + \left(\alpha \times T^b \times (1+x)^a\right)\right)^a\right) + \\
&\left(\begin{array}{l}
\alpha \times \left(T + \left(p_v \times (1+x)^a\right) + \left(\alpha \times T^b (1+x)^a\right) + (\gamma \times x)\right)^b \times \\
\left(1 + x + \left(p_v (1+x)^a\right) + \left(\alpha \times T^b \times (1+x)^a\right)\right)^a -
\end{array}\right) \\
&\left(\begin{array}{l}
\alpha\left(T + \left(p_u (1+x)^a\right) + \left(\alpha \times T^b \times (1+x)^a\right) + (\gamma \times x)\right)^b \times \\
\left(1 + x + \left(p_u \times (1+x)^a\right) + \left(\alpha \times T^b (1+x)^a\right)\right)^a
\end{array}\right) + \left(\gamma \times (p_v - p_u)\right).
\end{aligned}
\tag{3.1}
$$

Substituting, $y = \left(\alpha \times T^b \times (1 + x^a)\right)$, $w = (1 + x)^a$ and $\lambda = \frac{p_v}{p_u}$ we obtain:

$$C_u(\pi') - C_v(\pi) = [w \times ((\lambda \times p_u) - p_u)] + [p_u \times (1 + x + (\lambda \times p_u \times w) + y)^a] - [\lambda \times p_u (1 + x + (p_u \times w) + y)^a] + \Big[\alpha \times (T + (\lambda p_u \times w) + y + (\gamma \times x))^b \times (1 + x + (\lambda \times p_u \times w) + y)^a\Big] - [\alpha \times (T + (p_u \times w) + y + (\gamma \times x))^b \times (1 + x + (p_u \times w) + y)^a] + [\gamma \times ((\lambda \times p_u) - p_u)]. \tag{3.2}$$

From Lemma A.1, we have

$$\begin{bmatrix} C_u(\pi') - C_v(\pi) = ((\lambda \times p_u) - p_u) + (p_u \times ((\lambda \times t) + 1)^a) - \\ ((\lambda \times p_u) \times (t+1)^a) + \left(\alpha (t+1)^a ((P_u \times \lambda \times x^a) + y)^b\right) - \\ \left(\alpha ((\lambda \times t) + 1)^a ((P_u \times x^a) + y)^b\right) \end{bmatrix} \geq 0.$$

Consequently, $C_u(\pi') > C_v(\pi)$. □

The makespan under π is strictly less than that of π'. This contradicts the optimally of π'.

Theorem 3.2. *The problem* $1\left|\left[\left[p_r + \left(\alpha \times t_r^b\right)\right]\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right] + s_r^{\mathrm{psd}}\right| \sum C_j$ *can be solved optimally by sequencing jobs in non-decreasing order of their processing times (SPT rule).*

Proof. Consider an optimal schedule π, which contains two adjacent jobs, job J_u followed by job J_v $(v = u + 1)$, such that $p_u < p_v$. t is total completion time of all jobs before J_u when the starting time of J_u is T and C_u and C_v express completion time of J_u and J_v, scheduled at position u and $(v = u + 1)$, respectively.

$$\sum C(\pi) = T + 2 \times \left[\left(\left(p_u + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right) + \gamma \sum_{k=1}^{r-1} p_k\right] + \left[\begin{pmatrix} \left(p_v + \left(\alpha\left[T + \left(\left(p_u + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right) + \gamma \sum_{k=1}^{r-1} p_k\right]^b\right)\right) \times \\ \left(1 + \sum_{k=1}^{r-1} p_k + \left(\left(p_u + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right)\right)^a + \end{pmatrix} \\ \gamma \sum_{k=1}^{r-1} p_k + \gamma\left(\left(p_u + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right) \right].$$

By interchanging jobs J_u and J_v, we obtain schedule π' where the starting time of J_v is T. The completion times of the jobs processed before jobs J_u and J_v

are not affected by interchange and, therefore:

$$\sum C(\pi') = T + 2 \times \left[\left(\left(p_v + \left(\alpha \times T^b\right)\right) \left(1 + \sum_{k=1}^{r-1} p_k\right)^a \right) + \gamma \sum_{k=1}^{r-1} p_k \right] + \left[\begin{array}{l} \left(\begin{array}{l} \left(p_u + \left(\alpha \times \left[T + \left(\left(p_v + \left(\alpha \times T^b\right)\right) \left(1 + \sum_{k=1}^{r-1} p_k\right)^a \right) + \gamma \sum_{k=1}^{r-1} p_k \right]^b \right) \right) \times \\ \left(1 + \sum_{k=1}^{r-1} p_k + \left(\left(p_v + \left(\alpha \times T^b\right)\right) \left(1 + \sum_{k=1}^{r-1} p_k\right)^a \right) \right)^a \end{array} \right) + \\ \gamma \sum_{k=1}^{r-1} p_k + \gamma \left(\left(p_v + \left(\alpha \times T^b\right)\right) \left(1 + \sum_{k=1}^{r-1} p_k\right)^a \right) \end{array} \right].$$

Again, substituting $x = \sum_{k=1}^{r-1} p_k$ yields that the difference between the values of $\sum C(\pi)$ and $\sum C(\pi')$ is:

$$\begin{array}{l} \sum C(\pi') - \sum C(\pi) = \left(2 \times (1+x)^a (p_v - p_u)\right) + \\ \left(p_u \times \left(1 + x + \left(p_v \times (1+x)^a\right) + \left(\alpha \times T^b (1+x)^a\right)\right)^a\right) - \\ \left(p_v \times \left(1 + x + \left(p_u (1+x)^a\right) + \left(\alpha \times T^b \times (1+x)^a\right)\right)^a\right) + \\ \left(\begin{array}{l} \alpha \times \left(T + \left(p_v \times (1+x)^a\right) + \left(\alpha \times T^b \times (1+x)^a\right) + (\gamma \times x)\right)^b \times \\ \left(1 + x + \left(p_v (1+x)^a\right) + \left(\alpha \times T^b \times (1+x)^a\right)\right)^a \end{array} \right) - \\ \left(\begin{array}{l} \alpha \times \left(T + \left(p_u \times (1+x)^a\right) + \left(\alpha \times T^b \times (1+x)^a\right) + (\gamma \times x)\right)^b \times \\ \left(1 + x + \left(p_u \times (1+x)^a\right) + \left(\alpha \times T^b \times (1+x)^a\right)\right)^a \end{array} \right) + \\ \left(\gamma \times (p_v - p_u)\right). \end{array} \tag{3.3}$$

Substituting, $y = \left(\alpha \times T^b \times (1 + x^a)\right)$, $w = (1+x)^a$ and $\lambda = \frac{p_v}{p_u}$ we obtain:

$$\begin{array}{c} \sum C(\pi') - \sum C(\pi) = \left[2 \times w \times ((\lambda \times p_u) - p_u)\right] + \\ \left[p_u \times (1 + x + (\lambda \times p_u \times w) + y)^a\right] - \left[\lambda \times p_u \times (1 + x + (p_u \times w) + y)^a\right] + \\ \left[\alpha \times (T + (\lambda \times p_u \times w) + y + (\gamma \times x))^b (1 + x + (\lambda \times p_u \times w) + y)^a\right] - \\ \left[\alpha \times (T + (p_u \times w) + y + (\gamma \times x))^b (1 + x + (p_u \times w) + y)^a\right] + \\ \left[\gamma \times ((\lambda \times p_u) - p_u)\right] \end{array} \tag{3.4}$$

From Lemma A.1 it follows that:

$$\left[\begin{array}{l} \sum C(\pi') - \sum C(\pi) = [2 \times w \times ((\lambda \times p_u) - p_u)] + \\ [p_u \times (1 + x + (\lambda \times p_u \times w) + y)^a] - [\lambda \times p_u \times (1 + x + (p_u \times w) + y)^a] + \\ \left[\alpha \times (T + (\lambda \times p_u \times w) + y + (\gamma \times x))^b \times (1 + x + (\lambda \times p_u \times w) + y)^a\right] - \\ \left[\alpha \times (T + (p_u \times w) + y + (\gamma \times x))^b (1 + x + (p_u \times w) + y)^a\right] + \\ [\gamma \times ((\lambda \times p_u) - p_u)] \end{array}\right] > 0$$

$$\sum C(\pi') > \sum C(\pi).$$

π dominates π', which contradicts the optimally of π'. □

Townsed [18] studied the single machine scheduling with quadratic objectives. Wang [15] examined problem $1\|C_j^2$ with the effects of the learning and deterioration, showing that the problem can be solved optimally by sequencing jobs in non-decreasing order of their processing times with the effects of the learning and deterioration. We can show that the SPT sequence still holds for the problem $1\left|\left[\left[p_r + \left(\alpha \times t_r^b\right)\right]\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right] + s_r^{psd}\right|\sum C_j^2$.

Theorem 3.3. *The problem* $1\left|\left[\left[p_r + \left(\alpha \times t_r^b\right)\right]\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right] + s_r^{psd}\right|\sum C_j^2$ *can be solved optimally by sequencing jobs in non-decreasing order of their processing times (SPT rule).*

Proof. Follows directly from Theorem 3.2. Since $C_u(\pi') > C_v(\pi)$ and $C_v(\pi') > C_u(\pi)$, it clearly follows that $C_u^2(\pi') > C_v^2(\pi)$ and $C_v^2(\pi') > C_u^2(\pi)$. □

Theorem 3.4. *For the problem* $1\left|\left[\left[p_r + \left(\alpha \times t_r^b\right)\right]\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right] + s_r^{psd}\right|L_{\max}$, *if jobs have agreeable due dates, i.e.* $p_u < p_v$ *implies* $d_u < d_v$ *for all jobs* J_u *and* J_v, *an optimal schedule can be solved optimally by sequencing the jobs in non-decreasing order of* d_j *(EDD rule).*

Proof. Consider an optimal schedule π which contains two adjacent jobs, job J_u followed by job J_v $(v = u + 1)$, such that $p_u < p_v$. The starting time of J_u is T and L_u and L_v express the lateness of J_u and J_v, scheduled at positions u and

$(v = u + 1)$, respectively. The lateness of this job pair is:

$$L_u(\pi) = T + \left[\left(\left(p_u + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right) + \gamma \sum_{k=1}^{r-1} p_k\right] - d_u$$

$$L_v(\pi) = \left[\left[\begin{array}{l}\left(\left(p_v + \left(\alpha \times \left[T + \left(\left(p_u + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right) + \gamma \sum_{k=1}^{r-1} \hat{p}_k\right]^b\right)\right)\right. \\ \left.\times \left(1 + \sum_{k=1}^{r-1} p_k + \left(\left(p_u + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right)\right)^a\right) \\ + \gamma \sum_{k=1}^{r-1} p_k + \gamma \times \left(\left(p_u + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right)\end{array}\right]\right]$$

$$-d_v$$

$$L_v(\pi) = C_v(\pi) - d_v.$$ □

Performing a pairwise interchange on jobs J_u and $J_v$yields schedule π' for which the starting time of J_v is T. The completion times of the jobs processed before jobs J_u and J_v are not affected by interchange and the lateness of the job pair is now:

$$L_v(\pi') = T + \left[\left(\left(p_v + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right) + \gamma \sum_{k=1}^{r-1} p_k\right] - d_v$$

$$L_u(\pi') = \left[\left[\begin{array}{l}\left(\left(p_u + \left(\alpha \times \left[T + \left(\left(p_v + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right) + \gamma \sum_{k=1}^{r-1} p_k\right]^b\right)\right)\right. \\ \left.\times \left(1 + \sum_{k=1}^{r-1} p_k + \left(\left(p_v + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right)\right)^a\right) \\ + \gamma \sum_{k=1}^{r-1} p_k + \gamma \times \left(\left(p_v + \left(\alpha \times T^b\right)\right)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right)\end{array}\right]\right]$$

$$-d_u$$

$$L_v(\pi') = C_u(\pi) - d_u.$$

The difference between the values of $L_v(\pi')$ and $L_u(\pi)$ is

$$L_v(\pi') - L_u(\pi) = \left((p_v - p_u)\left(1 + \sum_{k=1}^{r-1} p_k\right)^a\right) + (d_u - d_v). \tag{3.5}$$

Furthermore difference between the values of $L_u(\pi')$ and $L_v(\pi)$ is

$$L_u(\pi') - L_v(\pi) = (C_u(\pi') - C_v(\pi)) + (d_v - d_u). \tag{3.6}$$

It follows from Theorem 3.1 that $C_u(\pi') > C_v(\pi)$ when $p_u < p_v$, $a < 0$ and $b > 0$. Furthermore, $L_v(\pi') - L_u(\pi) < 0$ and $L_u(\pi') - L_v(\pi) > 0$ are obtained using Equation (3.5) and Equation (3.6) when $d_u < d_v$.

π dominates π', which contradicts the optimally of π'.

4. Conclusions

This paper considers several single machine problems with past sequence dependent setup times under the simultaneous effect of non-linear deterioration and exponential learning. We show that the makespan, the sum of completion times and the sum of completion times square are minimize by sequencing jobs according to the SPT rule. The problem for minimizing the maximum lateness with agreeable due dates is shown to be solved by the EDD rule.

Acknowledgements. M. Duran TOKSARI's research was supported by the The Scientific and Technological Research Council of Turkey (TÜBÝTAK).

A. Appendix

Lemma A.1.

$$\left[\begin{array}{l} [w \times ((\lambda \times p_u) - p_u)] + [p_u\,(1 + x + (\lambda \times p_u \times w) + y)^a] - \\ [\lambda \times p_u \times (1 + x + (p_u \times w) + y)^a] + \\ [\alpha \times (T + (\lambda \times p_u \times w) + y + (\gamma \times x))^b \times (1 + x + (\lambda \times p_u \times w) + y)^a] - \\ \left[\alpha \times (T + (p_u \times w) + y + (\gamma \times x))^b \times (1 + x + (p_u w) + y)^a\right] + \\ [\gamma\,((\lambda p_u) - p_u)] \end{array}\right] > 0$$

when $(T, w, \alpha, b, x, y, \gamma > 0)$, $(a < 0)$, and $(\lambda \geq 1)$.

Proof.

$$f(\lambda) = \left[\begin{array}{l} [w \times ((\lambda \times p_u) - p_u)] + [p_u \times (1 + x + (\lambda \times p_u \times w) + y)^a] - \\ [\lambda \times p_u\,(1 + x + (p_u \times w) + y)^a] + \\ \left[\alpha \times (T + (\lambda \times p_u \times w) + y + (\gamma x))^b\,(1 + x + (\lambda \times p_u \times w) + y)^a\right] - \\ \left[\alpha \times (T + (p_u \times w) + y + (\gamma \times x))^b \times (1 + x + (p_u \times w) + y)^a\right] + \\ [\gamma \times ((\lambda \times p_u) - p_u)] \end{array}\right].$$

Taking the first derivative of $f(\lambda)$ with respect to λ, we have

$$f'(\lambda) = \left[\begin{array}{l} [w \times \lambda] + \left[(p_u)^2 \times a \times w \times (1 + x + (\lambda \times p_u \times w) + y)^{a-1}\right] - \\ [p_u \times (1 + x + (p_u \times w) + y)^a] + \\ \left[\begin{array}{l} \alpha \times \left(p_u \times w \times b \times (T + (\lambda \times p_u \times w) + y + (\gamma \times x))^{b-1} \times (1 + x + (\lambda \times p_u \times w) + y)^a\right) + \\ \left(p_u \times w \times a \times (T + (\lambda \times p_u \times w) + y + (\gamma \times x))^b \times (1 + x + (\lambda \times p_u \times w) + y)^{a-1}\right) \end{array}\right] \\ + [\gamma \times \lambda] \end{array}\right]$$

and

$$f''(\lambda) = \begin{bmatrix} [w] + \left[(p_u)^3\, a \times (a-1) \times w^2 \times (1 + x + (\lambda \times p_u \times w) + y)^{a-2}\right] + \\ \begin{bmatrix} \alpha \times p_u \times w \times b \times \begin{pmatrix} p_u \times w \times (b-1) \times (T + (\lambda \times p_u \times w) + y + (\gamma \times x))^{b-2} \times \\ (1 + x + (\lambda \times p_u \times w) + y)^a \end{pmatrix} + \\ \left(p_u \times w \times a \times (T + (\lambda \times p_u \times w) + y + (\gamma \times x))^{b-1} (1 + x + (\lambda \times p_u \times w) + y)^{a-1}\right) \end{bmatrix} + \\ \begin{bmatrix} \alpha \times p_u \times w \times a \begin{pmatrix} p_u \times w \times b \times (T + (\lambda \times p_u \times w) + y + (\gamma \times x))^{b-1} \times \\ (1 + x + (\lambda \times p_u \times w) + y)^{a-1} \end{pmatrix} + \\ \left(p_u \times w \times (a-1) \times (T + (\lambda \times p_u \times w) + y + (\gamma \times x))^b (1 + x + (\lambda \times p_u \times w) + y)^{a-2}\right) \end{bmatrix} \\ \times [\gamma] \end{bmatrix}$$

Hence, $f'(\lambda)$ is increasing when $(T, w, \alpha, b, x, y, \gamma > 0)$, $(a < 0)$, and $(\lambda \geq 1)$ for $f''(\lambda) \geq 0$.

Hence, $f(\lambda)$ is increasing when $(T, w, \alpha, b, x, y, \gamma > 0)$, $(a < 0)$, and $(\lambda \geq 1)$. Therefore, we have

$$\begin{bmatrix} [w \times ((\lambda \times p_u) - p_u)] + [p_u \times (1 + x + (\lambda \times p_u \times w) + y)^a] - \\ [\lambda \times p_u \times (1 + x + (p_u \times w) + y)^a] + \\ \left[\alpha \times (T + (\lambda \times p_u \times w) + y + (\gamma \times x))^b (1 + x + (\lambda \times p_u \times w) + y)^a\right] - \\ \left[\alpha \times (T + (p_u \times w) + y + (\gamma \times x))^b (1 + x + (p_u \times w) + y)^a\right] + \\ [\gamma \times ((\lambda \times p_u) - p_u)] \end{bmatrix} > 0$$

for $(T, w, \alpha, b, x, y, \gamma > 0)$, $(a < 0)$, and $(\lambda \geq 1)$.

This completes the proof. □

References

[1] B. Alidaee and N.K. Womer, Scheduling with time dependent processing times: Review and extensions. *J. Oper. Res. Soc.* **50** (1999) 711–720.

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

[3] S. Browne and U. Yechiali, Scheduling deteriorating jobs on a single processor. *Oper. Res.* **38** (1990) 495–498.

[4] T.C.E. Cheng, Q. Ding and B.M.T. Lin, A concise survey on the scheduling problems with deteriorating processing times. *Eur. J. Oper. Res.* **152** (2003) 1–13.

[5] T.C.E. Cheng, Chin-Chia Wu and Wen-Chiung Lee, Some scheduling problems with deteriorating jobs and learning effects. *Comp. Ind. Eng.* **54** (2008) 972–982.

[6] T. Eren and E. Güner, Minimizing total tardiness in a scheduling problem with a learning effect. *Appl. Math. Model.* **31** (2007) 1351–1361.

[7] J.N.D. Gupta and S.K. Gupta, Single facility scheduling with nonlinear processing times. *Comput. Ind. Eng.* **14** (1988) 387–393.

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

[9] W.H. Kuo and D.L. Yang, Single machine group scheduling with past-sequence-dependent setup times and learning effects. *Inf. Process. Lett.* **102** (2007) 22–26.

[10] W.H. Kuo and D.L. Yang, Single machine group scheduling with a time dependent learning effect. *Eur. J. Oper. Res.* **33** (2006) 2099–2112.

[11] 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.

[12] G. Mosheiov, Scheduling problems with a learning effect. *Eur. J. Oper. Res.* **132** (2001) 687–693.

[13] G. Mosheiov and J.B. Sidney, Scheduling with general job-dependent learning curves. *Eur. J. Oper. Res.* **147** (2003) 665–670.

[14] M.D. Toksari and E. Guner, Minimizing the earliness/tardiness costs on parallel machine with learning effects and deteriorating jobs: a mixed nonlinear integer programming approach. *Adv. Manuf. Technol.* in press.

[15] W. Townsend, The single machine problem with quadratic penalty function of completion times: a branch-and-bound solution. *Manage. Sci.* **24** (1978) 530–534.

[16] G.T. Voutsinas and C.P. Pappis, Scheduling jobs with values exponentially deteriorating over time. *Int. J. Prod. Econ.* **79** (2002) 163–169.

[17] X. Wang and T.C.E. Cheng, Single-machine scheduling with deteriorating jobs and learning effects to minimize the makespan. *Eur. J. Oper. Res.* **178** (2007) 57–70.

[18] J.B. Wang, Single-machine scheduling problems with the effects of learning and deterioration. *Omega* **35** (2007) 397–402.