Single-machine Scheduling with Position-based and Sum-of-processing-time-based Learning Effects

Wen-Chiung Lee and Chin-Chia Wu

Abstract -
The learning effects are widely studied in scheduling problems recently. Biskup [1] classified the learning effect models into two diverse approaches, namely the position -based and the sum-of-processing-time-based learning effect models. He further claimed that the position-based learning effect model seems to be a realistic assumption for the case that the actual processing of the job is mainly machine-driven, while the sum-of-processing-time -based learning effect model takes into account the effects that the human gains from the experience of operating similar tasks. In this paper, we propose a learning model which considers both the machine and human learning effects simultaneously. We will show that the position-based learning and the sum-of- processing-time-based learning effect models are special cases of the proposed model. Moreover, we will derive the optimal solutions for some single-machine problems.

***Keywords*:** scheduling; learning effect; single-machine; flowshop

1. Introduction

Conventionally, it is assumed that the job processing times are fixed and known from the first job to be processed to the last job to be completed. However, this assumption might not be realistic in many situations. Biskup [2] claimed that unit costs decline as firms produce more of a product and gain knowledge or experience in several industrial empirical studies. For instance, repeated processing of similar tasks improves worker skills; workers are able to perform setup, to deal with machine operations and software, or to handle raw materials and components at a greater pace. This phenomenon is known as the "learning effect" in the literature.

The learning effect is relatively unexplored in the scheduling fields until recently although it has been investigated in a variety of industries. Biskup [2] was the first author to bring the concept of learning effect into scheduling problems. He proposed a position-based learning model and showed some single-machine scheduling problems remain polynomially solvable. Since then, many researchers have paid more attention on the relatively young but very vivid area. Mosheiov [3] showed that solving scheduling problems with a learning effect requires more computational effort than that required for solving the original problem. Mosheiov [4] further considered the flow-time problem on parallel identical machines. Lee *et al.* [5] studied a single-machine bi-criterion scheduling problem. Zhao *et al.* [6] developed the optimal solutions for some single machine and flowshop problems under some special cases. Lee and Wu [7] considered the two-machine flowshop total completion problem, while Chen *et al.* [8] considered a bi-criterion flowshop problem. Wang and Xia [9] demonstrated that Johnson's rule for the two-machine flowshop makespan problem does not necessarily lead to an optimal schedule if the learning effect is present. Wu *et al.* [10] considered the maximum lateness problem. Eren and Güner [11] addressed the single-machine total tardiness problem. Wu and Lee [12] considered single-machine scheduling with learning effect and an availability constraint. Lee and Wu [13] presented the optimal solutions for some single-machine group scheduling problems. Wu and Lee [14] studied the multiple-machine flowshop total completion time problem. Other extensions of the position-based learning models can be found in Cheng and Wang [15], Bachman and Janiak [16], Wang [17] and Wang [18].

On the other hand, Kuo and Yang [19] proposed a sum-of-processing-time- based learning effect model. He showed that the makespan and the total completion time problems are polynomially solvable. Other form of the sum-of-processing-time- based learning model can be found in Koulamas and Kyparisis [20]. Wu and Lee [21] studied some single-machine scheduling problems with the sum-of -processing-time-based learning effect. Recently, Biskup [1] provided an extensive review of scheduling with learning effects. He claimed that the position-based learning effect model assumes that learning takes place by processing time independent operations like setting up the machines. This seems to be a realistic assumption for the case that the actual processing time of the job is mainly machine-driven and has (near to) none human interference. The sum-of-processing -time approach takes into account the experience the workers gain from producing the jobs. This might, for example, be the case for offset printing, where running the press itself is a highly complicated and error-prone process. In many realistic situations, both the machine and human learning effects might exist simultaneously. In this paper, we propose a new model which considers both the human and the machine learning effects at the same time.

The remainder of this paper is organized as follows. The problem formulation is given in the next section. In Section 3, the solution procedures for some single-machine problems are presented. The conclusion is given in the last section.

Manuscript received November 20, 2008. This paper is support by NSC under grant number NSC 97-2221- E-035-070

Wen-Chiung Lee is with Feng Chia University, Taichung, Taiwan, ROC. (phone:886-4-2456-7250x4016; e-mail: wclee@fcu.edu.tw).

Chin-Chia Wu is with Feng Chia University, Taichung, Taiwan, ROC. (e-mail: cchwu@fcu.edu.tw).

2. Problem formulation

Formulation of the proposed learning effect model is as follows. There are n jobs ready to be processed on a single machine. Each job *j* has a normal processing time p_j and a due date d_j. Due to the learning effect, the actual processing time of job *j* is

$$p_{j[r]} = p_j r^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]})^{a_2},$$

if it is scheduled in the *r*th position in a sequence, where $p_{[l]}$ is the normal processing time of the job scheduled in the *l*th position in the sequence, $c_0 > 0$ is a constant, r is the scheduled position, $a_1 \le 0$ and $a_2 \le 0$ are the learning indices**.** For convenience, we denote the learning model by *LE* [3]. For instance, the single-machine makespan problem is denoted as $1|LE|C_{\max}$ using conventional notation (Graham *et al.* [22]) for describing scheduling problems. This scheduling model unifies the job-position-based learning effect and the sum-of- processing-time-based learning effect models. For instance, it is the position-based learning effect model $p_{j[r]} = p_j r^{a_1}$ if $a_2 = 0$, while it is the sum-of-processing -time-based learning effect model

$$p_{j[r]} = p_j(1+p_{[1]}+...+p_{[r-1]})^{a_2} \text{ if } a_1 = 0.$$

3. Some single-machine problems

In this section, solutions of several single-machine problems under the proposed learning model are developed. Before presenting the main results, we first state the lemmas that will be used in the proofs of the properties in the sequel. The proofs are given in the appendix.

Lemma 1. $1+acx(1+x)^{a-1}-c(1+x)^a \ge 0$ for $a<0$, $0<c<1$ and $x \ge 0$.

Lemma 2. $\lambda-1+c(1+\lambda x)^a - \lambda c(1+x)^a \ge 0$ for $\lambda \ge 1$, $a<0$, $0<c<1$ and $x \ge 0$.

Lemma 3. $1+k[1-c(1+x)^a]+acx(1+kx)^{a-1} \ge 0$ for $k \ge 1$, $a<0$, $0<c<1$, and $x \ge 0$.

Lemma 4. $k[1-c(1+x)^a]-[1-c(1+kx)^a]/k \ge 0$ for $k \ge 1$, $a<0$, $0<c<1$, and $x \ge 0$.

Lemma 5.
$(\lambda-1)+\lambda k[1-c(1+x)^a]-[1-c(1+\lambda kx)^a]/k \ge 0$ for $\lambda \ge 1$, $k \ge 1$, $a<0$, $0<c<1$, and $x \ge 0$.

We will prove the properties using the well-known pairwise interchange technique. Suppose that S and S' are two job schedules and the difference between S and S' is a pairwise interchange of two adjacent jobs *i* and *j*, i.e., $S=(\pi,i,j,\pi')$ and $S'=(\pi,j,i,\pi')$, where π and π' each denote a partial sequence. Furthermore, we assume that there are *r*-1 scheduled jobs in π and A is the completion time of the last job in π. Under *S*, the completion times of jobs *i* and *j* are respectively

$$C_i(S) = A + p_i r^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]})^{a_2} \quad (1)$$

and

$$C_j(S) = A + p_i r^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]})^{a_2} + p_j(r+1)^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]}+p_i)^{a_2}. \quad (2)$$

Similarly, the completion times of jobs *j* and *i* in S' are respectively

$$C_j(S') = A + p_j r^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]})^{a_2} \quad (3)$$

and

$$C_i(S') = A + p_j r^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]})^{a_2} + p_i(r+1)^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]}+p_j)^{a_2}. \quad (4)$$

Property 1. For the $1|LE|C_{\max}$ problem, the optimal schedule is obtained by sequencing jobs in the shortest processing time (SPT) order.

Proof: Suppose that $p_i \le p_j$. To show that S dominates S', it suffices to show that $C_j(S) \le C_i(S')$.

Taking the difference between Equations (2) and (4), we have

$$\begin{aligned} C_i(S')-C_j(S) = {} & p_j r^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]})^{a_2} \\ & + p_i(r+1)^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]}+p_j)^{a_2} \\ & - p_i r^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]})^{a_2} \\ & - p_j(r+1)^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]}+p_i)^{a_2}. \end{aligned} \quad (5)$$

Substituting $\lambda = p_j/p_i$, $c=(1+1/r)^{a_1}$, and $x = p_i/(1+\sum_{l=1}^{r-1} p_{[l]})$ into Equation (5), it is derived from Lemma 1 that

$$C_i(S')-C_j(S) = p_i r^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]})^{a_2}\{\lambda-1+c(1+\lambda x)^{a_2}-\lambda c(1+x)^{a_2}\} \ge 0$$

since $\lambda = p_j/p_i \ge 1$, $0<c<1$, $x \ge 0$, $a_1 \le 0$ and $a_2 \le 0$. Thus, S dominates S'. Therefore, repeating this interchange argument for jobs not sequenced in the SPT order completes the proof of the property.

Property 2. For the $1|LE|\sum C_j$ problem, the optimal schedule is obtained by sequencing jobs in the SPT order.

Proof: The proof is omitted since it is similar to that of Property 1.

The well-known weighted smallest processing time (WSPT) rule provides the optimal schedule for the classical total weighted completion time problem. However, the WSPT order does not yield an optimal schedule under the proposed learning model, as shown by the example below.

Example 1. Given $n=2$, $p_1=3$, $p_2=2$, $w_1=2$, $w_2=1$, $a_1=-0.322$ and $a_2=-1$. The WSPT sequence (1, 2) yields a total weighted completion time of 9.4, while the sequence (2, 1) yields the optimal value of 7.6.

Although the WSPT order does not provide an optimal schedule under the proposed learning model, it still gives an optimal solution if the processing times and the weights are agreeable, i.e., $p_j / p_i \geq w_j / w_i \geq 1$ for all jobs *i* and *j*. The result is stated in the following theorem.

Property 3. For the $1|LE|\sum w_j C_j$ problem, the optimal schedule is obtained by sequencing jobs in non-decreasing order of p_i / w_i if the processing times and the weights are agreeable.

Proof: Suppose that $p_j / p_i \geq w_j / w_i \geq 1$. Since $p_i \leq p_j$, it implies from Property 1 that $C_j(S) \leq C_i(S')$. Thus, to show that S dominates S', it suffices to show that $w_i C_i(S) + w_j C_j(S) \leq w_j C_j(S') + w_i C_i(S')$. From Equations (1) to (4), we have

$$\begin{aligned}&[w_j C_j(S') + w_i C_i(S')] - [w_i C_i(S) + w_j C_j(S)]\\ &= (w_i p_j - w_j p_i) r^{a_1} (1+\sum_{l=1}^{r-1} p_{[l]})^{a_2}\\ &\quad + w_j p_j [r^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]})^{a_2} - (r+1)^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]} + p_i)^{a_2}]\\ &\quad - w_i p_i [r^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]})^{a_2} - (r+1)^{a_1}(1+\sum_{l=1}^{r-1} p_{[l]} + p_j)^{a_2}]. \end{aligned} \tag{6}$$

Substituting $\lambda = p_j w_i / p_i w_j$, $k = w_j / w_i$, $c = (1+1/r)^{a_1}$, and $x = p_i / (1+\sum_{l=1}^{r-1} p_{[l]})$ into Equation (6), we have from Lemma 5 that

$$\begin{aligned}&[w_j C_j(S') + w_i C_i(S')] - [w_i C_i(S) + w_j C_j(S)]\\ &= w_j p_i r^{a_1} (1+\sum_{l=1}^{r-1} p_{[l]})^{a_2} \{(\lambda - 1)\\ &\quad + \lambda k [1 - c(1+x)^{a_2}] - \frac{1}{k}[1 - c(1+\lambda k x)^{a_2}]\} \geq 0, \end{aligned}$$

since $\lambda = p_j w_i / p_i w_j \geq 1$, $k \geq 1$, $0 < c < 1$, $x \geq 0$, $a_1 \leq 0$ and $a_2 \leq 0$. Thus, S dominates S'. Repeating this interchange argument for jobs not sequenced in the WSPT order completes the proof of Property 3.

Let $L_i = C_i - d_i$ denote the lateness of job *i*, for *i* = 1, 2,…, *n*. Ordering jobs according to the earliest due-date (EDD) rule provides the optimal sequence for the classical maximum lateness problem. However, this policy is not optimal under the proposed learning model, as shown by the example below.

Example 2. Given $n=2$, $p_1=3$, $p_2=2$, $d_1=3$, $d_2=4$, $a_1=-0.322$ and $a_2=-1$. The EDD sequence (1, 2) yields a maximum lateness of 0, while the sequence (2, 1) yields the optimal value of -0.2.

Although the EDD order does not provide the optimal solution for the maximum lateness problem under the proposed model, it is still optimal if the job processing times and the due dates are agreeable, i.e., $d_i \leq d_j$ implies $p_i \leq p_j$ for all jobs *i* and *j*. The result is stated in the following theorem.

Property 4. For the $1|LE|L\max$ problem, the optimal schedule is obtained by sequencing jobs in non-decreasing order of d_i if the job processing times and the due dates are agreeable.

Proof: Suppose that $d_i \leq d_j$. This implies that $p_i \leq p_j$. Thus, it is seen from Theorem 1 that $C_j(S) \leq C_i(S')$. To show that S dominates S', it suffices to show that $\max\{L_i(S), L_j(S)\} \leq \max\{L_i(S'), L_j(S')\}$. By definition, the lateness of jobs *i* and *j* in *S* and jobs *j* and *i* in S' are respectively

$L_i(S) = C_i(S) - d_i$,

$L_j(S) = C_j(S) - d_j$,

$L_j(S') = C_j(S') - d_j$,

and

$L_i(S') = C_i(S') - d_i$.

Since $p_i \leq p_j$, we have from Property 1 that

$$C_j(S) \leq C_i(S'). \tag{7}$$

With the condition that $d_i \leq d_j$, we have

$$L_j(S) \leq L_i(S'). \tag{8}$$

From equation (7), and since job *i* is processed before job *j* in *S*, we have

$$L_i(S) \leq L_i(S'). \tag{9}$$

From equations (8) and (9), we have

$$\max\{L_i(S), L_j(S)\} \leq \max\{L_i(S'), L_j(S')\}.$$

Thus, repeating this interchange argument for all the jobs not sequenced in the EDD rule completes the proof of Property 4.

In the following, we will show the EDD rule provides the optimal solution for the total tardiness problem if the job processing times and the due dates are agreeable, i.e., $d_i \leq d_j$ implies $p_i \leq p_j$ for all jobs *i* and *j*.

Property 5. For the $1|LE|\sum T_i$ problem, the optimal schedule is obtained by sequencing jobs in non-decreasing order of d_i if the job processing times and the due dates are agreeable.

Proof: Suppose that $d_i \leq d_j$. It also implies $p_i \leq p_j$. The total tardiness of the first *r*-1 jobs are the same since they are processed in the same order. Since the makespan is minimized by the SPT rule (Property 1), the total tardiness of partial sequence π' in *S* will not be greater than that of partial sequence π' in S'. Thus, to prove that the total tardiness of *S* is less than or equal to that of S', it suffices to show that $T_i(S) + T_j(S) \leq T_j(S') + T_i(S')$.

From Equations (1) - (4), it is derived that the tardiness of jobs *i* and *j* in *S* are

$$T_i(S) = \max\{A + p_i r^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]})^{a_2} - d_i, 0\},$$

and

$$T_j(S) = \max\{A + p_i r^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]})^{a_2} + p_j (r+1)^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]} + p_i)^{a_2} - d_j, 0\}.$$

Similarly, the tardiness of jobs *i* and *j* in S' are

$$T_j(S') = \max\{A + p_j r^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]})^{a_2} - d_j, 0\},$$

and

$$T_i(S') = \max\{A + p_j r^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]})^{a_2} + p_i (r+1)^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]} + p_j)^{a_2} - d_i, 0\}.$$

To compare the total tardiness of jobs *i* and *j* in *S* and in S', we divide it into two cases. In the first case that $A + p_j r^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]})^{a_2} \le d_j$, the total tardiness of jobs *i* and *j* in *S* and in S' are

$$T_i(S) + T_j(S) = \max\{A + p_i r^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]})^{a_2} - d_i, 0\} + \max\{A + p_i r^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]})^{a_2} + p_j (r+1)^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]} + p_i)^{a_2} - d_j, 0\},$$

and

$$T_j(S') + T_i(S') = \max\{A + p_j r^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]})^{a_2} + p_i (r+1)^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]} + p_j)^{a_2} - d_i, 0\}.$$

Suppose that neither $T_i(S)$ nor $T_j(S)$ is zero. Note that this is the most restrictive case since it comprises the case that either one or both $T_i(S)$ and $T_j(S)$ are zero. From Property 1 and $d_i \le d_j$, we have

$$\{T_j(S') + T_i(S')\} - \{T_i(S) + T_j(S)\} = (p_j - p_i) r^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]})^{a_2} + p_i (r+1)^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]} + p_j)^{a_2} - p_j (r+1)^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]} + p_i)^{a_2} + d_j - p_i r^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]})^{a_2} - A \ge 0.$$

Thus, $\{T_j(S') + T_i(S')\} - \{T_i(S) + T_j(S)\} \ge 0$ in the first case. In the second case that $A + p_j r^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]})^{a_2} > d_j$, the total tardiness of jobs *i* and *j* in *S* and in S' are

$$T_i(S) + T_j(S) = \max\{A + p_i r^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]})^{a_2} - d_i, 0\} + \max\{A + p_i r^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]})^{a_2} + p_j (r+1)^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]} + p_i)^{a_2} - d_j, 0\},$$

and

$$T_j(S') + T_i(S') = 2A + 2 p_j r^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]})^{a_2} + p_i (r+1)^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]} + p_j)^{a_2} - d_i - d_j.$$

Suppose that neither $T_i(S)$ nor $T_j(S)$ is zero. From Property 1 and $p_i \le p_j$, we have

$$\{T_j(S') + T_i(S')\} - \{T_i(S) + T_j(S)\} = 2(p_j - p_i) r^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]})^{a_2} + p_i (r+1)^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]} + p_j)^{a_2} - p_j (r+1)^{a_1} (1 + \sum_{l=1}^{r-1} p_{[l]} + p_i)^{a_2} \ge 0.$$

Thus, $\{T_j(S') + T_i(S')\} - \{T_i(S) + T_j(S)\} \ge 0$ in the second case. This completes the proof of Property 5.

4. Conclusions

In many realistic situations, the learning effects of machines and humans exist simultaneously. In this paper, we propose a new learning model which considers the position- based learning effect and the sum-of-processing-time-based learning effect at the same time. We derived polynomial-time optimal solutions for two single-machine problems as to minimize the makespan and the total completion time. In addition, we show that the problems to minimize the total weighted completion time, the maximum lateness, and the total tardiness are polynomially solvable under certain agreeable conditions.

References

[1] D. Biskup, “A state-of-the-art review on scheduling with learning effect.” *European Journal of Operational Research*, 188, 2008, pp. 315-329.
[2] D. Biskup, “Single-machine scheduling with learning considerations.” *European Journal of Operational Research*, 115, 1999, pp. 173–178.
[3] G. Mosheiov, “Scheduling problems with a learning effect.” *European Journal of Operational Research*, 132, 2001, pp. 687–693.
[4] G. Mosheiov, “Parallel machine scheduling with a learning effect.” *Journal of the Operational Research Society*, 52, 2001, pp. 1–5.
[5] W. C. Lee, C. C. Wu and H. J. Sung, “A bi-criterion single-machine scheduling problem with learning considerations.” *Acta Informatica*, 40, 2004, pp. 303–315.
[6] C. L. Zhao, Q. L. Zhang and H. Y. Tang, “Machine scheduling problems with learning effects.” *Dynamics of Continuous, Discrete and Impulsive Systems, Series A: Mathematical Analysis*, 11, 2004, pp. 741–747.
[7] W. C. Lee and C. C. Wu, “Minimizing total completion time in a two-machine flowshop with a learning effect.” *International Journal of Production Economics*, 88, 2004, pp. 85–93.
[8] P. Chen, C. C. Wu and W. C. Lee, “A bi-criteria two-machine flowshop scheduling problem with a learning effect.” *Journal of the Operational Research Society*, 57, 2006, pp. 113–1125.

[9] J. B. Wang and Z. Q. Xia, "Flow-shop scheduling with a learning effect." *Journal of the Operational Research Society,* 56, 2005, pp. 1325–1330.
[10] C. C. Wu, W. C. Lee and T. C. Chen, "Heuristic algorithms for solving the maximum lateness scheduling problem with learning considerations." *Computers and Industrial Engineering,* 52, 2007, pp. 124-132.
[11] T. Eren and E. Guner, "Minimizing total tardiness in a scheduling problem with a learning effect." *Applied Mathematical Modelling,* 31, 2007, pp. 1351–1361.
[12] C. C. Wu and W. C. Lee, "A note on single-machine scheduling with learning effect and an availability constraint." *International Journal of Advanced Manufacturing Technology* 33, 2007, pp. 540-544.
[13] W. C. Lee and C. C. Wu, "A note on single-machine group scheduling problems with position-based learning effect." *Applied Mathematical Modelling*, 2008, (doi:10.1016/j.apm.2008.05.020).
[14] C. C. Wu and W. C. Lee, "A note on the total completion time problem in a permutation flowshop with a learning effect." *European Journal of Operational Research* 192, 2009, pp. 343-347.
[15] T. C. E. Cheng and G. Wang, "Single machine scheduling with learning effect considerations." *Annals of Operations Research*, 98, 2000, pp. 273–290.
[16] A. Bachman and A. Janiak, "Scheduling jobs with position-dependent processing times." *Journal of the Operational Research Society,* 55, 2004, pp. 257–264.
[17] J. B. Wang, "A note on scheduling problems with learning effects and deteriorating jobs." *International Journal of Systems Science*, 37, 2006, pp. 827–833.
[18] J. B. Wang, "Single-machine scheduling problems with the effects of learning and deterioration." *Omega, 35*, 2007, pp. 397–402.
[19] W.. Kuo and D.. Yang, "Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect." *European Journal of Operational Research,* 174, 2006, pp. 1184–1190.
[20] C. Koulamas and G. J. Kyparisis, "Single-machine and two-machine flowshop scheduling with general learning function." *European Journal of Operational Research*, 178, 2007, pp. 402–407.
[21] C. C. Wu and W. C. Lee, "Single-machine scheduling problems with a learning effect." *Applied Mathematical Modelling* 32, 2008, pp. 1191-1197.
[22] 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, pp. 287-326.

Appendix

Lemma 1. $1+acx(1+x)^{a-1}-c(1+x)^a \ge 0$ for $a<0$, $0<c<1$ and $x \ge 0$.

Proof: Let $f(x)=1+acx(1+x)^{a-1}-c(1+x)^a$. Taking the first derivative of $f(x)$ with respect to *x*, we have

$$f'(x)=a(a-1)cx(1+x)^{a-2} \ge 0$$

for $a<0$, $0<c<1$ and $x \ge 0$. Thus, this implies that $f(x)$ is a non-decreasing function on $x \ge 0$. Since $f(0)=1-c>0$ for $a<0$ and $0<c<1$, we have

$$f(x) \ge 0$$

for $a<0$, $0<c<1$ and $x \ge 0$. This completes the proof.

Lemma 2. $\lambda-1+c(1+\lambda x)^a-\lambda c(1+x)^a \ge 0$ for $\lambda \ge 1$, $a<0$, $0<c<1$ and $x \ge 0$.

Proof: Let $g(\lambda)=\lambda-1+c(1+\lambda x)^a-\lambda c(1+x)^a$. Taking the first and second derivatives of $g(\lambda)$ with respect to λ, we have

$$g'(\lambda)=1+acx(1+\lambda x)^{a-1}-c(1+x)^a$$

and

$$g''(\lambda)=a(a-1)cx^2(1+\lambda x)^{a-2}.$$

Since $\lambda \ge 1$, $a<0$, $0<c<1$ and $x \ge 0$, it implies that $g''(\lambda) \ge 0$. Therefore, $g'(\lambda)$ is a non-decreasing function for $\lambda \ge 1$. From Lemma 1, we have

$$g'(1)=1+acx(1+x)^{a-1}-c(1+x)^a \ge 0.$$

Using the fact that $g'(\lambda)$ is a non-decreasing function for $\lambda \ge 1$, this implies that

$$g'(\lambda) \ge g'(1) \ge 0.$$

Therefore, it also implies that $g(\lambda)$ is a non-decreasing function for $\lambda \ge 1$. Since $g(1)=0$, we have

$$g(\lambda) \ge 0$$

for $\lambda \ge 1$, $a<0$, $0<c<1$ and $x \ge 0$. This completes the proof.

Lemma 3. $1+k[1-c(1+x)^a]+acx(1+kx)^{a-1} \ge 0$ for $k \ge 1$, $a<0$, $0<c<1$, and $x \ge 0$.

Proof: Let $f(x)=1+k[1-c(1+x)^a]+acx(1+kx)^{a-1}$. Taking the first derivative of $f(x)$ with respect to x, we have

$$f'(x)=-kac(1+x)^{a-1}+ac(1+kx)^{a-1}$$
$$+a(a-1)ckx(1+kx)^{a-2}.$$

Since $k \ge 1$, $a<0$, and $x \ge 0$, we have $(1+x)^{a-1}>(1+kx)^{a-1}$. Thus, $f'(x)>0$. This implies that $f(x)$ is a non-decreasing function for $x \ge 0$. Since $f(0)=1+k[1-c]>0$, we have $f(x)>0$. This completes the proof.

Lemma 4. $k[1-c(1+x)^a]-[1-c(1+kx)^a]/k \geq 0$ for $k \geq 1$, $a < 0$, $0 < c < 1$, and $x \geq 0$.

Proof: Let $f(x) = k[1-c(1+x)^a]-[1-c(1+kx)^a]/k$. Taking the first derivative of $f(x)$ with respect to x, we have

$f'(x) = -kac(1+x)^{a-1} + ac(1+kx)^{a-1}$.

Since $k \geq 1$, $a < 0$, $0 < c < 1$, $x \geq 0$, we have $(1+x)^{a-1} > (1+kx)^{a-1}$. Thus, $f'(x) > 0$. This implies that $f(x)$ is a non-decreasing function for $k \geq 1$, $a < 0$, $0 < c < 1$, and $x \geq 0$. Therefore, $f(x) \geq f(0) = (k-1/k)(1-c) > 0$. This completes the proof.

Lemma 5.
$(\lambda-1)+\lambda k[1-c(1+x)^a]-[1-c(1+\lambda kx)^a]/k \geq 0$ for $\lambda \geq 1$, $k \geq 1$, $a < 0$, $0 < c < 1$, and $x \geq 0$.

Proof:
Let $g(\lambda) = (\lambda-1)+\lambda k[1-c(1+x)^a]-[1-c(1+\lambda kx)^a]/k$. Taking the first and second derivatives of $g(\lambda)$ with respect to λ, we have

$g'(\lambda) = 1 + k[1-c(1+x)^a] + acx(1+\lambda kx)^{a-1}$,

and

$g''(\lambda) = a(a-1)ckx^2(1+\lambda kx)^{a-2}$.

Since $\lambda \geq 1$, $k \geq 1$, $a < 0$, $0 < c < 1$, and $x \geq 0$, we have $g''(\lambda) \geq 0$. This implies that $g'(\lambda)$ is a non-decreasing function for $\lambda \geq 1$. From Lemma 3, we have

$g'(\lambda) \geq g'(1) = 1 + k[1-c(1+x)^a] + acx(1+kx)^{a-1} \geq 0$.

This implies that $g'(\lambda) \geq 0$ and $g(\lambda)$ is a non-decreasing function for $\lambda \geq 1$, too. Therefore, we have from Lemma 4 that

$g(\lambda) \geq g(1) = k[1-c(1+x)^a]-[1-c(1+kx)^a]/k \geq 0$.

This completes the proof.