Some scheduling problems with sum-of-processing-times-based and job-position-based learning effects

T.C.E. Cheng*[,a], Chin-Chia Wu[b], and Wen-Chiung Lee[b]

[a]*Department of Logistics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong*

[b]*Department of Statistics, Feng Chia University, Taichung, Taiwan*

10 February 2008

Abstract

In this paper we introduce a new scheduling model with learning effects in which the actual processing time of a job is a function of the total normal processing times of the jobs already processed and of the job's scheduled position. We show that the single-machine problems to minimize makespan and total completion time are polynomially solvable. In addition, we show that the problems to minimize total weighted completion time and maximum lateness are polynomially solvable under certain agreeable conditions. Finally, we present polynomial-time optimal solutions for some special cases of the *m*-machine flowshop problems to minimize makespan and total completion time.

Keywords: Scheduling; learning effect; single-machine; flowshop

---

*Corresponding author. E-mail: LGTcheng@polyu.edu.hk

1. Introduction

Learning effects in scheduling problems have recently received growing attention from the scheduling research community. For instance, Biskup [1] pointed out that repeated processing of similar tasks improves workers' skills, e.g., workers are able to perform setups, deal with machine operations or software, or handle raw materials and components at a faster pace. Heizer and Render [9], and Russell and Taylor [18] demonstrated through empirical studies in several industries that unit costs decline as firms produce more of a product and gain knowledge or experience. Besides, Lee *et al*. [12] utilized the genetic learning mechanism to elicit participants' meeting behavior. Chen and Hsiang [5] pointed out that corporations have felt the pressure of fast-paced innovations and knowledge transfer as major driving forces in raising their sustainable competitive advantage and organizational total productivity. Thus, the development of the knowledge community through e-learning is an important strategy in implementing knowledge management policy.

The impact of learning on productivity in manufacturing in the aircraft industry was first discovered by Wright [25], and it was subsequently confirmed in many industries in both the manufacturing and service sectors [27]. Although the learning effect had been investigated in a variety of industries, it had not been unexplored in the field of scheduling until Biskup's [1] study. He introduced a scheduling model with learning effects in which the actual processing time of a job is a function of its position in the schedule. He showed that the single-machine problems to minimize total deviations of job completion times from a common due date and to minimize the sum of job completion times are polynomially solvable. Mosheiov [15] found that under Biskup's learning effect model the optimal schedules for some classical scheduling problems remain valid, but they require much greater computational effort to obtain. Mosheiov and Sidney [16] extended the model in which the learning effects gained from doing some jobs are stronger than those from the other jobs, i.e., the so-called job-dependent learning model. Lee *et al.* [13] studied a bi-criterion scheduling problem on a single machine. Chen *et al*. [3] considered a two-machine flowshop scheduling problem in which the objective is to minimize the weighted sum of total completion time and maximum tardiness. Eren and Güner [6] applied the 0-1 integer programming approach to derive the optimal solution, and used a random search, the tabu search, and simulated annealing to obtain near-optimal solutions for the total tardiness problem. Cheng *et al.* [4] considered some permutation flowshop scheduling problems with learning effects on no-idle dominant machines. Eren and Güner [7] further considered a bi-criterion flowshop scheduling problem.

Besides the job-position-based learning effect scheduling model, there are several other learning effect scheduling models in the literature. For instance, Wang and Cheng [21] considered a single-machine scheduling problem with a volume-dependent, piecewise linear processing time function to model the learning effect. They showed that the maximum lateness problem is NP-hard in the strong sense and identified two special cases that are polynomially solvable. Wang and Xia [24] studied the flowshop problems when the learning effect is present. They provided the worst-case bound for the shortest processing time (SPT) rule for the problems to minimize makespan and total completion time. They also showed that the problems remain polynomially solvable for two special cases. Wang [22] studied a model in which the job processing times are functions of their starting times and positions in the sequence. Recently, Koulamas and Kyparisis [11] introduced a general sum-of-job-processing-times-based learning effect scheduling model in which employees learn more if they perform a job with a longer processing time. Bikup [2] provided a comprehensive review of the scheduling models and problems with considerations of the learning effect. Moreover, he gave detailed descriptions of the models and the authors' works. Janiak and Rudek [10] proposed a new approach to studying the learning effect in scheduling. Wang *et al.* [23] studied several single-machine scheduling problems with a time-dependent learning effect. Xu *et al.* [26] provided the worst case analysis for some flowshop problems.

Biskup [2] classified learning models into two types, namely position-based learning and sum-of-processing-time-based learning. He further claimed that position-based learning assumes that learning takes place by processing time independent operations like setting up of machines. This seems to be a realistic assumption for the case where the actual processing time of a job is mainly machine-driven and has (or is near to) none human interference. On the other hand, sum-of-processing-time-based learning takes into account the experience workers have gained from producing the same or similar jobs over time. This might, for example, be the case for offset printing, where running the press itself is a highly complicated and error-prone process.

In this paper we study a new learning effect scheduling model in which the learning effects of machines and humans are both present, i.e., the actual processing time of a job is a function not only of the total normal processing times of the jobs already processed, but also of the job's scheduled position. The remainder of this paper is organized as follows. The solution procedures for the single-machine problems to minimize makespan, total completion time, total weighted completion time, and maximum lateness minimization under the proposed model are presented in the next section. In Section 3 we consider special cases of the flowshop problems to minimize makespan and total completion time. We conclude the paper in the last section.

2. Some single-machine problems

Suppose that there are n jobs to be scheduled on a single machine. Each job i

has a normal processing time p_i and a due date d_i. Specially, the actual processing time of job *j* if it is scheduled in the *r*th position in a sequence is

$$p_{j[r]} = p_j\left(1-\frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2},$$

where p_j is the basic processing time of job *j*, $p_{[l]}$ denotes the basic processing time of the job scheduled in the *l*th position in the sequence, and a_1 and a_2 denote two learning indices with $a_1 \geq 1$ and $a_2 < 0$. This learning effect scheduling model is adapted from the sum-of-processing-time-based learning effect scheduling model by Koulamas and Kyparisis [11] and the job-position-based learning effect scheduling model by Biskup [1]. Under the proposed model, the actual processing time of a job depends not only on the job's scheduled position, but also on the processing times of the jobs already processed. It is observed from the model that the longer the jobs that have already been processed or the later a job is scheduled, the stronger the learning effect is on the subsequent jobs that are yet to be processed.

In this section we will discuss several single-machine problems under the proposed learning model. The makespan and the total completion time problems are shown to be polynomially solvable, while the total weighted completion time and the maximum lateness problems are shown to be polynomially solvable under certain agreeable conditions. Suppose that *S* and S' are two job schedules. 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 jobs in π. Thus, jobs *i* and *j* are the *r*th and (*r*+1)th job in *S*, whereas jobs *j* and *i* are scheduled in the *r*th and (*r*+1)th position in S'. In addition, let B denote the completion time of the last job in π. Under *S*, the completion times of jobs *i* and *j* are respectively

$$C_i(S) = B + p_i \left(1 - \frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l} \right)^{a_1} r^{a_2}, \tag{1}$$

and

$$C_j(S) = B + p_i \left(1 - \frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l} \right)^{a_1} r^{a_2} + p_j \left(1 - \frac{\sum_{l=1}^{r-1} p_{[l]} + p_i}{\sum_{l=1}^{n} p_l} \right)^{a_1} (r+1)^{a_2}. \tag{2}$$

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

$$C_j(S') = B + p_j \left(1 - \frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l} \right)^{a_1} r^{a_2}, \tag{3}$$

and

$$C_i(S') = B + p_j \left(1 - \frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l} \right)^{a_1} r^{a_2} + p_i \left(1 - \frac{\sum_{l=1}^{r-1} p_{[l]} + p_j}{\sum_{l=1}^{n} p_l} \right)^{a_1} (r+1)^{a_2}. \tag{4}$$

Theorem 1. For the makespan problem under the proposed learning model, the optimal schedule is obtained by sequencing the jobs in the 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

$$C_i(S') - C_j(S) = (p_j - p_i) \left(1 - \frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l} \right)^{a_1} r^{a_2} + p_i \left(1 - \frac{\sum_{l=1}^{r-1} p_{[l]} + p_j}{\sum_{l=1}^{n} p_l} \right)^{a_1} (r+1)^{a_2}$$

$$-\,p_j\left(1-\frac{\sum_{l=1}^{r-1}p_{[l]}+p_i}{\sum_{l=1}^{n}p_l}\right)^{a_1}(r+1)^{a_2}\quad . \tag{5}$$

Substituting $t=\left(1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l}\right)$, $y=\frac{p_i}{\sum_{l=1}^{n}p_l}$, $\lambda=\frac{p_j}{p_i}$, and $x=\frac{y}{t}$ into equation (5), we have

$$C_i(S')-C_j(S)=p_i t^{a_1} r^{a_2}\{\lambda[1-(1-x)^{a_1}(\frac{r+1}{r})^{a_2}]-[1-(1-\lambda x)^{a_1}(\frac{r+1}{r})^{a_2}]\}. \tag{6}$$

Since $\lambda=\frac{p_j}{p_i}\geq 1$, we have from Lemma 2 that

$$C_i(S')-C_j(S)\geq 0. \tag{7}$$

Thus, S dominates S'. Therefore, repeating this interchange argument for all the jobs not sequenced in the SPT order completes the proof of the theorem.

Theorem 2. For the total completion time problem under the proposed learning model, the optimal schedule is obtained by sequencing jobs in the SPT order.

Proof. Suppose that $p_i \leq p_j$. From Theorem 1, we have $C_j(S)<C_i(S')$ since $p_i \leq p_j$. To show that S dominates S', it suffices to show that $C_i(S)+C_j(S)\leq C_j(S')+C_i(S')$.

From equations (1) to (4), we have

$\{C_j(S')+C_i(S')\}-\{C_i(S)+C_j(S)\}$

$$=(p_j-p_i)\left(1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l}\right)^{a_1} r^{a_2}+(p_j-p_i)\left(1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l}\right)^{a_1} r^{a_2}$$

$$+\, p_i \left(1-\frac{\sum_{l=1}^{r-1} p_{[l]} + p_j}{\sum_{l=1}^{n} p_l}\right)^{a_1} (r+1)^{a_2} - p_j \left(1-\frac{\sum_{l=1}^{r-1} p_{[l]} + p_i}{\sum_{l=1}^{n} p_l}\right)^{a_1} (r+1)^{a_2}. \qquad (8)$$

The first term of equation (8) is non-negative since $p_i \le p_j$. It is also noted from the proof of Theorem 1 that the sum of the last three terms of equation (8) is also non-negative. This implies that $C_i(S)+C_j(S) \le C_j(S')+C_i(S')$. Thus, repeating this interchange argument for all the jobs not sequenced in the SPT rule completes the proof of Theorem 2.

Smith [19] showed that sequencing jobs according to the weighted smallest processing time (WSPT) rule provides an optimal schedule for the classical total weighted completion time problem, i.e., sequencing jobs in non-decreasing order of p_j / w_j, where w_j is the weight of job *j*. However, the WSPT order does not yield an optimal schedule under the proposed learning model, as shown by the example below.

Example 1. Given $n = 2$, $p_1 = 3$, $p_2 = 2$, $w_1 = 2$, $w_2 = 1$, $a_1 = 2$ and $a_2 = -0.322$. The WSPT sequence (1, 2) yields a total weighted completion time of 9.256, while the sequence (2, 1) yields the optimal value of 7.728.

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., $\frac{p_j}{p_i} \ge \frac{w_j}{w_i} \ge 1$ for all jobs *i* and *j*. The result is stated in the following theorem.

Theorem 3. For the total weighted completion time problem under the proposed learning model, an 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, i.e., $\frac{p_j}{p_i} \ge \frac{w_j}{w_i} \ge 1$ for all jobs *i* and *j*.

Proof. Suppose that $\frac{p_j}{p_i} \ge \frac{w_j}{w_i} \ge 1$. Since $p_i \le p_j$, it is seen from Theorem 1 that $C_j(S) \le C_i(S')$. Thus, to show that S dominates S', it suffices to show that $w_iC_i(S) + w_jC_j(S) \le w_jC_j(S') + w_iC_i(S')$. From equations (1) to (4), we have

$$[w_jC_j(S') + w_iC_i(S')] - [w_iC_i(S) + w_jC_j(S)]$$

$$= \{w_j[B + p_j(1 - \frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} r^{a_2}] + w_i[B + p_j(1 - \frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} r^{a_2} + p_i(1 - \frac{\sum_{l=1}^{r-1} p_{[l]} + p_j}{\sum_{l=1}^{n} p_l})^{a_1} (r+1)^{a_2}]\}$$

$$-\{w_i[B + p_i(1 - \frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} r^{a_2}] + w_j[B + p_i(1 - \frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} r^{a_2} + p_j(1 - \frac{\sum_{l=1}^{r-1} p_{[l]} + p_i}{\sum_{l=1}^{n} p_l})^{a_1} (r+1)^{a_2}]\}$$

$$= (w_i p_j - w_j p_i)(1 - \frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} r^{a_2} \quad + w_j p_j[(1 - \frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} r^{a_2} - (1 - \frac{\sum_{l=1}^{r-1} p_{[l]} + p_i}{\sum_{l=1}^{n} p_l})^{a_1} (r+1)^{a_2}]$$

$$-w_i p_i[(1 - \frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} r^{a_2} - (1 - \frac{\sum_{l=1}^{r-1} p_{[l]} + p_j}{\sum_{l=1}^{n} p_l})^{a_1} (r+1)^{a_2}]. \qquad (9)$$

Substituting $\lambda = \frac{p_j / w_j}{p_i / w_i} \ge 1$, $t = (1 - \frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l})$, $x = \frac{p_i}{\sum_{l=1}^{n} p_l - \sum_{l=1}^{r-1} p_{[l]}}$, and $k = \frac{w_j}{w_i}$ into equation (9), we have

$$[w_jC_j(S') + w_iC_i(S')] - [w_iC_i(S) + w_jC_j(S)]$$

$$= w_j p_i t^{a_1} r^{a_2} \{(\lambda - 1) + \lambda k[1 - (1-x)^{a_1} (\frac{r+1}{r})^{a_2}] - \frac{1}{k}[1 - (1 - \lambda k x)^{a_1} (\frac{r+1}{r})^{a_2}]\}. \qquad (10)$$

From Lemma 5, the value of equation (10) is non-negative, so we have

$$w_jC_j(S') + w_iC_i(S') \ge w_iC_i(S) + w_jC_j(S).$$

Thus, repeating this interchange argument for all the jobs not sequenced in the WSPT order completes the proof of Theorem 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 $n = 2$ **,** $p_1 = 20$, $p_2 = 30$, $d_1 = 30$, $d_2 = 28$, $a_1 = 2$ and $a_2 = -0.322$. The EDD sequence (2, 1) yields a maximum lateness of 2.56, while the sequence (1, 2) yields the optimal value of 0.64.

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., $d_i \le d_j$ implies $p_i \le p_j$ for all jobs *i* and *j*. The result is stated in the following theorem.

Theorem 4. For the maximum lateness problem under the proposed learning model, an optimal schedule is obtained by sequencing jobs in non-decreasing order of d_i (i.e., the EDD order) if the job processing times and the due dates are agreeable, i.e., $d_i \le d_j$ implies $p_i \le p_j$ for all jobs *i* and *j*.

Proof. Suppose that $d_i \le d_j$. This implies that $p_i \le p_j$. Thus, it is seen from Theorem 1 that $C_j(S) \le C_i(S')$. To show that *S* dominates S', it suffices to show that $\max\{L_i(S), L_j(S)\} \le \max\{L_j(S'), L_i(S')\}$. 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 $p_i \le p_j$, we have from Theorem 1 that

$$C_j(S) \le C_i(S'). \tag{11}$$

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

$$L_j(S) \le L_i(S'). \tag{12}$$

From equation (11), and since job *i* is processed before job *j* in *S*, we have

$$L_i(S) \le L_i(S'). \tag{13}$$

From equations (12) and (13), we have

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

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

3. Two flowshop problems

In most manufacturing environments, a set of processes is sequentially performed in several stages to complete a job. Such a system is referred to as the flow shop environment [8, 14, 17, 20]. Formulation of the flowshop scheduling problem is described as follows. Suppose that there is a set of *n* jobs, say $N = \{1, 2, \ldots, n\}$, to be processed on *m* machines $M_1, M_2, \ldots, M_m$. Each job *j* consists of *m* operations O_{1j}, O_{2j}, …, O_{mj}, where O_{ij} must be processed on machine M_i, $i = 1, 2, \ldots, m$. Processing of operation $O_{i+1, j}$ can start only after O_{ij} has been completed. A machine can handle one job at a time and preemption is not allowed. Moreover, we only consider a permutation schedule. The actual processing time p_{ijr} of job j on machine M_i if it is scheduled in the rth position in a sequence is

$$p_{ij[r]} = p_{ij}(1-\frac{\sum_{l=1}^{r-1} p_{i[l]}}{\sum_{l=1}^{n} p_{il}})^{a_1} r^{a_2} ,$$

where p_{ij} is the basic processing time of job j on machine i, $p_{i[l]}$ denotes the basic processing time of the job scheduled in the lth position on machine i in the sequence, a_1 and a_2 denote two learning indices with $a_1 \geq 1$ and $a_2 < 0$. For a given schedule π, $C_{ij} = C_{ij}(\pi)$ represents the completion time of operation O_{ij}, and $C_j = C_{mj}$ denotes the completion time of job j. In this paper we consider the special case where the processing times on all the machines for any given job are identical, i.e., $p_{ij} = p_j$. For the traditional flowshop problem, Pinedo [17] showed that the completion time of the jth job in a given sequence S is as follows:

$$C_{[j]}(S) = \sum_{k=1}^{j} p_{[k]} + (m-1)\max\{p_{[1]}, p_{[2]}, ..., p_{[j]}\} .$$

where $p_{[k]}$ is the basic common processing time of the job scheduled in the kth position in the sequence. Similarly, the completion time of the jth job in a given sequence S under the proposed learning model is

$$C_{[j]}(S) = \sum_{k=1}^{j} (1-\frac{\sum_{l=1}^{k-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} k^{a_2} p_{[k]}$$

$$+(m-1)\max\{p_{[1]}, (1-\frac{p_{[1]}}{\sum_{l=1}^{n} p_l})^{a_1} 2^{a_2} p_{[2]}, ..., (1-\frac{\sum_{l=1}^{j-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} j^{a_2} p_{[j]}\} . \qquad (14)$$

Theorem 5. For the flowshop makespan problem under the proposed learning model, the optimal schedule is obtained by sequencing jobs in the SPT order if $p_{ij} = p_j$.

Proof. Suppose that S and S' are two job schedules. 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 jobs in π. Thus, jobs i and j are the rth and (r+1)th job in S, whereas jobs j and i are scheduled in the rth and (r+1)th position in S'. In addition, let B denote the completion time of the last job in π. Under S, the completion time of job j is

$$C_j(S)=B+p_i\left(1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l}\right)^{a_1} r^{a_2}+p_j\left(1-\frac{\sum_{l=1}^{r-1}p_{[l]}+p_i}{\sum_{l=1}^{n}p_l}\right)^{a_1}(r+1)^{a_2}+$$

$$(m-1)\max\{p_{[1]},\ldots,(1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l})^{a_1} r^{a_2} p_i\ ,(1-\frac{\sum_{l=1}^{r-1}p_{[l]}+p_i}{\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}\ p_j\}. \qquad (15)$$

Similarly, the completion time of job i in S' is

$$C_i(S')=B+p_j\left(1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l}\right)^{a_1} r^{a_2}+p_i\left(1-\frac{\sum_{l=1}^{r-1}p_{[l]}+p_j}{\sum_{l=1}^{n}p_l}\right)^{a_1}(r+1)^{a_2}+$$

$$(m-1)\max\{p_{[1]},\ldots,(1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l})^{a_1} r^{a_2} p_j\ ,(1-\frac{\sum_{l=1}^{r-1}p_{[l]}+p_j}{\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}\ p_i\}. \qquad (16)$$

Suppose that $p_i \le p_j$. To show that S dominates S', it suffices to show that $C_j(S)\le C_i(S')$. Since $p_i \le p_j$, $a_1 \ge 1$, and $a_2 \le 0$, we have

$$(1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l})^{a_1} r^{a_2} p_j \ge (1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l})^{a_1} r^{a_2} p_i\ ,$$

and

$$(1-\frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} r^{a_2} p_j \geq (1-\frac{\sum_{l=1}^{r-1} p_{[l]} + p_i}{\sum_{l=1}^{n} p_l})^{a_1} (r+1)^{a_2} p_j .$$

This implies that

$$\max\{p_{[1]},\ldots,(1-\frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} r^{a_2} p_j ,(1-\frac{\sum_{l=1}^{r-1} p_{[l]} + p_j}{\sum_{l=1}^{n} p_l})^{a_1} (r+1)^{a_2} p_i\}$$

$$\geq \max\{p_{[1]},\ ,\ldots,(1-\frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} r^{a_2} p_i ,(1-\frac{\sum_{l=1}^{r-1} p_{[l]} + p_i}{\sum_{l=1}^{n} p_l})^{a_1} (r+1)^{a_2} p_j\} . \qquad (17)$$

From equations (15) and (16), we have

$$C_i(S') - C_j(S) = \{ (1-\frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} r^{a_2} p_j + (1-\frac{\sum_{l=1}^{r-1} p_{[l]} + p_j}{\sum_{l=1}^{n} p_l})^{a_1} (r+1)^{a_2} p_i$$

$$+(m-1)\max\{p_{[1]},\ldots,(1-\frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} r^{a_2} p_j ,(1-\frac{\sum_{l=1}^{r-1} p_{[l]} + p_j}{\sum_{l=1}^{n} p_l})^{a_1} (r+1)^{a_2} p_i\}\}$$

$$-\{ (1-\frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} r^{a_2} p_i + (1-\frac{\sum_{l=1}^{r-1} p_{[l]} + p_i}{\sum_{l=1}^{n} p_l})^{a_1} (r+1)^{a_2} p_j$$

$$+(m-1)\max\{p_{[1]},\ldots,(1-\frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l})^{a_1} r^{a_2} p_i ,(1-\frac{\sum_{l=1}^{r-1} p_{[l]} + p_i}{\sum_{l=1}^{n} p_l})^{a_1} (r+1)^{a_2} p_j\}\} . \quad (18)$$

Substituting $t = 1-\frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l}$, $\lambda = \frac{p_j}{p_i}$, $y = \frac{p_i}{\sum_{l=1}^{n} p_l}$, and $x = \frac{y}{t}$ into equation (18), we have

$$C_i(S') - C_j(S) = t^{a_1} r^{a_2} p_i \{\lambda[1-(1-x)^{a_1} (\frac{r+1}{r})^{a_2}] - [1-(1-\lambda x)^{a_1} (\frac{r+1}{r})^{a_2}]\}$$

$$+(m-1)\times(\max\{p_{[1]},\ldots,(1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}p_j,(1-\frac{\sum_{l=1}^{r-1}p_{[l]}+p_j}{\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}p_i\}$$

$$-\max\{p_{[1]},,\ldots,(1-\frac{\sum_{l=1}^{r-1}p_{[l]}}{\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}p_i,(1-\frac{\sum_{l=1}^{r-1}p_{[l]}+p_i}{\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}p_j\}).$$

From equation (17) and Lemmas 1 and 2, we have

$$C_i(S')-C_j(S)\geq 0.$$

Thus, S dominates S'. Therefore, repeating this interchange argument for all the jobs not sequenced in the SPT order completes the proof of the theorem.

Theorem 6. For the flowshop total completion time problem under the proposed learning model, the optimal schedule is obtained by sequencing jobs in the SPT order if $p_{ij}=p_j$.

Proof. The proof is similar to that of Theorem 5 and so is omitted.

4. Conclusions

Scheduling problems with learning effects have captured many scheduling researchers' attention in recent years. However, most of the research assumed that the learning effects depend on job positions. Recently, a sum-of-processing-times-based learning effect has been proposed by Koulamas and Kyparisis [11]. In this paper we considered a new learning model in which the actual processing time of a job depends not only on the job's scheduled position, but also on the processing times of the jobs already processed. In particular, we showed that the single-machine makespan and total completion time problems are polynomially solvable under the proposed learning model. In addition, we showed that the total weighted completion time and the maximum lateness problems are polynomially solvable under certain agreeable conditions. Finally, we presented polynomial-time optimal solutions for two flowshop problems to minimize makespan and total completion time under the assumption of identical processing times on all the machines. For large-size problems, the normal processing time will be shortened significantly due to the two learning effects. Thus, to develop new learning models when both learning effects are present will be an interesting issue for future research.

Acknowledgments

We are thankful to two anonymous referees for their constructive comments on an earlier version of the paper. Cheng was supported in part by The Hong Kong Polytechnic under a grant from the *Area of Strategic Development in China Business Services*.

References

[1] D. Biskup, Single-machine scheduling with learning considerations, European Journal of Operational Research 115 (1999) 173-178.

[2] D. Biskup, A state-of-the-art review on scheduling with learning effect, European Journal of Operational Research (2007) (doi:10.1016/j.ejor.2007.05.040).

[3] P. Chen, C.C. Wu, W.C. Lee, A bi-criteria two-machine flowshop scheduling problem with a learning effect, Journal of the Operational Research Society 57 (2006) 1113-1125.

[4] M.B. Cheng, S.J. Sun, Y. Yu, A note on flow shop scheduling problems with a learning effect on no-idle dominant machines, Applied Mathematics and

Computation 184 (2007) 945-949.

[5] R.S. Chen, C.H. Hsiang, A study on the critical success factors for corporations embarking on knowledge community-based e-learning, Information Sciences 177 (2007) 570-586.

[6] T. Eren, E. Güner, Minimizing total tardiness in a scheduling problem with a learning effect, Applied Mathematical Modelling 31 (2007a) 1351-1361.

[7] T. Eren, E. Güner, A bicriteria flowshop scheduling with a learning effect, Applied Mathematical Modelling (2007b) (doi: 10.1016/j.apm.2007.06.009).

[8] J. Fondrevelle, A. Oulamara, M.C. Portmann, Permutation flowshop scheduling problems with time lags to minimize the weighted sum of machine completion times, International Journal of Production Economics 112 (2008) 168-176.

[9] J. Heiser, B. Render, Operations Management, 5th edition, Prentice Hall: Englewood Cliffs, NJ. (1999)

[10] A. Janiak, R. Rudek, A new approach to the learning effect: Beyond the learning curve restrictions, Computers and Operations Research (2007) (doi: 10.1016/j.cor.2007.04.007).

[11] C. Koulamas, G.J. Kyparisis, Single-machine and two-machine flowshop scheduling with general learning functions, European Journal of Operational Research 178 (2007) 402-407.

[12] C.S. Lee, C.C. Jiang, T.C. Hsieh, A genetic fuzzy agent using ontology model for meeting scheduling system, Information Sciences 176 (2006) 1131-1155.

[13] W.C. Lee, C.C. Wu, H.J. Sung, A bi-criterion single-machine scheduling problem with learning considerations, Acta Informatica 40 (2004) 303-315.

[14] B.M.T. Lin, C.Y. Lub, S.J. Shyuc, C.Y. Tsai, Development of new features of ant colony optimization for flowshop scheduling, International Journal of Production Economics (2007) (doi:10.1016/j.ijpe.2007.06.007).

[15] G. Mosheiov, Scheduling problems with a learning effect, European Journal of Operational Research 132 (2001) 687-693.

[16] G. Mosheiov, J.B. Sidney, Scheduling with general job-dependent learning curves, European Journal of Operational Research 147 (2003) 665-670.

[17] M. Pinedo, Scheduling: Theory, Algorithms, and Systems, 2nd edition, Prentice Hall: Englewood Cliffs, NJ (2002).

[18] R. Russell, B.W.III. Taylor, Operations Management: multimedia version, 3rd edition. Prentice Hall: Upper Saddle River, NJ (2000).

[19] W.E. Smith, Various optimizers for single state production, Naval Research Logistic Quarterly 3 (1956) 59-66.

[20] R. Tavakkoli-Moghaddam, A. Rahimi-Vahed, A.H. Mirzaei, A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: Weighted mean completion time and weighted mean tardiness, Information Sciences 177 (2007) 5072-5090.

[21] G.Q. Wang, T.C.E. Cheng, Single machine scheduling with learning effect considerations, Annals of Operations Research 98 (2000) 273-290.

[22] J.B. Wang, Single-machine scheduling problems with the effects of learning and deterioration, OMEGA-The International Journal of Management Science 35 (2007) 397-402.

[23] J.B. Wang, C.T. Ng, T.C.E. Cheng, L.L. Liu, Single-machine scheduling with a time-dependent learning effect, International Journal of Production Economics 111 (2008) 802-811.

[24] J.B. Wang, Z.Q. Xia, Flow-shop scheduling with a learning effect, Journal of the Operational Research Society 56 (2005) 1325-1330.

[25] T.P. Wright, Factors affecting the cost of airplanes, Journal of Aeronautical Science 3 (1936) 122-128.

[26] Z.Y. Xu, L.Y. Sun, J.T. Gong, Worst-case analysis for flow shop scheduling with a learning effect, International Journal of Production Economics (2007) (doi:10.1016/j.ijpe.2007.11.002).

[27] L.E. Yelle, The learning curve: historical review and comprehensive survey, Decision Science 10 (1979) 302-328.

Appendix

Lemma 1. $1-a_1x(1-x)^{a_1-1}(\frac{r+1}{r})^{a_2}-(1-x)^{a_1}(\frac{r+1}{r})^{a_2}\geq 0$ for $0\leq x\leq 1$, $a_1\geq 1$, $a_2<0$, and $r=1,2,...,n-1$.

Proof. Let $f(x)=1-a_1x(1-x)^{a_1-1}(\frac{r+1}{r})^{a_2}-(1-x)^{a_1}(\frac{r+1}{r})^{a_2}$. Taking the first derivative of $f(x)$ with respect to *x*, we have

$$f'(x)=-a_1(1-x)^{a_1-1}(\frac{r+1}{r})^{a_2}+a_1(a_1-1)x(1-x)^{a_1-2}(\frac{r+1}{r})^{a_2}+a_1(1-x)^{a_1-1}(\frac{r+1}{r})^{a_2}$$

$$=a_1(a_1-1)x(1-x)^{a_1-2}(\frac{r+1}{r})^{a_2}$$

$$\geq 0$$

for $0\leq x\leq 1$, $a_1\geq 1$, $a_2<0$, and $r=1,2,...,n-1$. Thus, this implies that $f(x)$ is a non-decreasing function on $0\leq x\leq 1$. Since $f(0)=1-(\frac{r+1}{r})^{a_2}>0$ for $a_2<0$ and $r=1,2,...,n-1$, we have

$$f(x)>0$$

for $0\leq x\leq 1$, $a_1\geq 1$, $a_2<0$, and $r=1,2,...,n-1$. This completes the proof.

Lemma 2. $\lambda[1-(1-x)^{a_1}(\frac{r+1}{r})^{a_2}]-[1-(1-\lambda x)^{a_1}(\frac{r+1}{r})^{a_2}]\geq 0$ for $1\leq\lambda\leq 1/x$, $0\leq x\leq 1$, $a_1\geq 1$, $a_2<0$, and $r=1,2,...,n-1$.

Proof. Let $g(\lambda)=\lambda[1-(1-x)^{a_1}(\frac{r+1}{r})^{a_2}]-[1-(1-\lambda x)^{a_1}(\frac{r+1}{r})^{a_2}]$. Taking the first and second derivatives of $g(\lambda)$ with respect to λ, we have

$$g'(\lambda)=1-(1-x)^{a_1}(\frac{r+1}{r})^{a_2}-a_1x(1-\lambda x)^{a_1-1}(\frac{r+1}{r})^{a_2}$$

and

$$g''(\lambda)=a_1(a_1-1)x^2(1-\lambda x)^{a_1-2}(\frac{r+1}{r})^{a_2}.$$

Since $a_1\geq 1$ and $1\leq\lambda\leq 1/x$, it implies that $g''(\lambda)\geq 0$. Therefore, $g'(\lambda)$ is a non-decreasing function for $1\leq\lambda\leq 1/x$. From Lemma 1, we have

$g'(1) = 1-(1-x)^{a_1}(\frac{r+1}{r})^{a_2} - a_1 x(1-x)^{a_1-1}(\frac{r+1}{r})^{a_2} > 0$.

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

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

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

$g(\lambda) \ge 0$

for $1 \le \lambda \le 1/x$, $0 \le x \le 1$, $a_1 \ge 1$, $a_2 < 0$, and $r = 1, 2, ..., n-1$. This completes the proof.

Lemma 3. $1 + k[1-(1-x)^{a_1}(\frac{r+1}{r})^{a_2}] - a_1 x(1-kx)^{a_1-1}(\frac{r+1}{r})^{a_2} \ge 0$ for $a_1 \ge 1$, $a_2 < 0$, $k \ge 1$, $0 < x < \frac{1}{k}$, and $r = 1, 2, \cdots, n-1$.

Proof. Let $f(x) = 1 + k[1-(1-x)^{a_1}(\frac{r+1}{r})^{a_2}] - a_1 x(1-kx)^{a_1-1}(\frac{r+1}{r})^{a_2}$. Taking the first derivative of $f(x)$ with respect to x, we have

$f'(x) = ka_1(1-x)^{a_1-1}(\frac{r+1}{r})^{a_2} - a_1(1-kx)^{a_1-1}(\frac{r+1}{r})^{a_2}$

$+ a_1(a_1-1)kx(1-kx)^{a_1-2}(\frac{r+1}{r})^{a_2}$.

Since $a_1 \ge 1$, $a_2 < 0$, $k \ge 1$, $0 < x < \frac{1}{k}$, and $r = 1, 2, \cdots, n-1$, we have $f'(x) > 0$. This implies that $f(x)$ is a non-decreasing function for $0 < x < \frac{1}{k}$. Since $f(0) = 1 + k[1-(\frac{r+1}{r})^{a_2}] > 0$, we have $f(x) > 0$. This completes the proof.

Lemma 4. $k[1-(1-x)^{a_1}(\frac{r+1}{r})^{a_2}] - \frac{1}{k}[1-(1-kx)^{a_1}(\frac{r+1}{r})^{a_2}] > 0$ for $a_1 \ge 1$, $a_2 < 0$, $k \ge 1$, $a_1 \ge 1, k \ge 1, 0 < x < \frac{1}{k}$, and $r = 1, 2, \cdots, n-1$.

Proof. Consider the following function

$$f(x) = k[1-(1-x)^{a_1}(\frac{r+1}{r})^{a_2}] - \frac{1}{k}[1-(1-kx)^{a_1}(\frac{r+1}{r})^{a_2}].$$

Taking the first derivative of $f(x)$ with respect to x, we have

$$f'(x) = ka_1(1-x)^{a_1-1}(\frac{r+1}{r})^{a_2} - a_1(1-kx)^{a_1-1}(\frac{r+1}{r})^{a_2}.$$

Since $a_1 \ge 1, k \ge 1, 0 < x < \frac{1}{k}$, and $k(1-x)^{a_1-1} - (1-kx)^{a_1-1} > 0$, we have $f'(x) > 0$.

This implies that $f(x)$ is a non-decreasing function for $a_1 \ge 1, k \ge 1, 0 < x < \frac{1}{k}$. Thus,

$$f(x) \ge f(0) = (k - \frac{1}{k})(1-(\frac{r+1}{r})^{a_2}) > 0.$$

This completes the proof.

Lemma 5. $(\lambda-1) + \lambda k[1-(1-x)^{a_1}(\frac{r+1}{r})^{a_2}] - \frac{1}{k}[1-(1-\lambda kx)^{a_1}(\frac{r+1}{r})^{a_2}] > 0$ for $a_1 \ge 1$, $a_2 < 0$, $k \ge 1$, $0 < x < 1$, $1 \le \lambda \le \frac{1}{kx}$, and $r = 1, 2, \cdots, n-1$.

Proof. Let $g(\lambda) = (\lambda-1) + \lambda k[1-(1-x)^{a_1}(\frac{r+1}{r})^{a_2}] - \frac{1}{k}[1-(1-\lambda kx)^{a_1}(\frac{r+1}{r})^{a_2}]$.

Taking the first and second derivatives of $g(\lambda)$ with respect to λ, we have

$$g'(\lambda) = 1 + k[1-(1-x)^{a_1}(\frac{r+1}{r})^{a_2}] - a_1 x[(1-\lambda kx)^{a_1-1}(\frac{r+1}{r})^{a_2}],$$

and

$$g''(\lambda) = a_1(a_1-1)kx^2(1-\lambda kx)^{a_1-2}(\frac{r+1}{r})^{a_2}.$$

Since $a_1 \ge 1$, $a_2 < 0$, $k \ge 1$, $0 < x < 1$, $1 \le \lambda \le \frac{1}{kx}$, and $r = 1, 2, \cdots, n-1$, we have $g''(\lambda) \ge 0$. This implies that $g'(\lambda)$ is a non-decreasing function for $1 \le \lambda \le \frac{1}{kx}$.

From Lemma 3, we have

$$g'(\lambda) \ge g'(1) = 1 + k[1-(1-x)^{a_1}(\frac{r+1}{r})^{a_2}] - a_1 x(1-kx)^{a_1-1}(\frac{r+1}{r})^{a_2} \ge 0.$$

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

$$g(\lambda) \geq g(1) = k[1-(1-x)^{a_1}(\frac{r+1}{r})^{a_2}] - \frac{1}{k}[1-(1-kx)^{a_1}](\frac{r+1}{r})^{a_2} \geq 0.$$

The proof is completed.