Some scheduling problems with deteriorating jobs and 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

Revised: 23 October 2007

Abstract

Although scheduling with deteriorating jobs and learning effect has been widely investigated, scheduling research has seldom considered the two phenomena simultaneously. However, job deterioration and learning co-exist in many realistic scheduling situations. In this paper we introduce a new scheduling model in which both job deterioration and learning exist simultaneously. The actual processing time of a job depends not only on the processing times of the jobs already processed but also on its scheduled position. For the single-machine case, we derive polynomial-time optimal solutions for the problems to minimize makespan and total completion time. In addition, we show that the problems to minimize total weighted completion time and maximum lateness are polynomially solvable under certain agreeable conditions. For the case of an *m*-machine permutation flowshop, we present polynomial-time optimal solutions for some special cases of the problems to minimize makespan and total completion time.

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

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

1. Introduction

Pinedo (2002) pointed out that sequencing and scheduling is a form of decision-making that plays a crucial role in manufacturing and service industries. In the current competitive business environment, effective sequencing and scheduling have become a necessity for survival in the marketplace. However, conventional scheduling models routinely assume that job processing times are known and fixed throughout the period of job processing. This assumption may be unrealistic in many situations since the processing times of jobs might be prolonged due to deterioration or shortened due to learning over time.

Kunnathur and Gupta (1990) pointed out that the temperature of an ingot, while waiting to enter a rolling machine, drops below a certain level, requiring the ingot to be reheated before rolling. Browne and Yechiali (1990) claimed that the time and effort required to put out a fire increase if there is a delay in the start of the fire-fighting effort. In such environments, a job that is processed later consumes more time than the same job if processed earlier. Scheduling in this setting is known as scheduling with deteriorating jobs, which was first independently introduced by Gupta and Gupta (1988) and Browne and Yechiali (1990). Since then, related models of scheduling with deteriorating jobs have been extensively studied from a variety of perspectives. Alidaee and Womer (1999) surveyed the rapidly growing literature, while Cheng *et al*. (2004) gave a detailed review of scheduling problems with deteriorating jobs.

On the other hand, Biskup (1999) 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 (1999) and Russell and Taylor (2000) demonstrated through empirical studies that unit costs decline as firms produce more of a product and gain knowledge or experience in several industries. The impact of learning on productivity improvement in manufacturing was first discovered in the aircraft industry by Wright (1936), and it was subsequently observed to exist in many other industries in both the manufacturing and service sectors (Yelle, 1979). Biskup (1999) and Cheng and Wang (2000) are among the pioneers that brought the concept of learning into the field of scheduling. Many researchers have since devoted much effort to studying this nascent and vivid area of scheduling with learning effects. Recently, Biskup (2007) discussed some of the economic fundamentals of scheduling and learning, and presented a comprehensive review of research of scheduling with learning effects.

Although the topics of deteriorating jobs and learning effect have been widely investigated in scheduling research recently, they have seldom been considered simultaneously. However, job deterioration and learning co-exist in many realistic scheduling situations. For example, Wang and Cheng (2007a) provided several real-life examples of processing environments involving task rotation where job deterioration is caused by forgetting effects, while the learning effect reflects that workers become more skilled to operate the machines through experience accumulation. Wang and Cheng (2007b) gave a practical example that the main stage in the production of porcelain crafts is to shape the raw material according to designs. Raw material, made up of clay and special coagulant, becomes harder with the lapse of time. It may result in increasing time to shape a craftwork. On the other hand, the productivity of the craftsmen can improve through increasing their proficiency in designs and operations. Wang (2007) pointed out that as the manufacturing environment becomes increasingly competitive, firms are moving towards shorter production runs and frequent product changes in order to offer faster services and provide customers with greater product varieties. The learning and forgetting that workers undergo in this environment have thus become increasingly important as workers tend to spend more time in rotating among tasks and responsibilities prior to becoming fully proficient in carrying out their operations. These workers are often interrupted by product and process changes that cause deterioration in their operational performance.

Lee (2004) showed that the single-machine problems to minimize makespan and total completion time are polynomially solvable under the learning and deteriorating scheduling models, in which the actual processing time of a job is $p_{jr} = \alpha_j t r^a$ or $p_{jr} = (p_0 + \alpha_j t) r^a$, where α_j is the rate of job deterioration, $t \geq 0$ is the starting time of processing the job, $a \leq 0$ is the learning index, and p_0 is the common basic processing time. Wang (2006) assumed that job processing times have the form: $p_{jr} = (\alpha_j + \beta t) r^a$, where α_j is the basic processing time and β is the common deteriorating rate. He showed that several single-machine and flowshop problems are polynomially solvable. In addition, Wang (2007) studied a model in which the job processing times have the form: $p_{jr} = p_j(\alpha(t) + \beta r^a)$, where p_j is the basic processing time and $\alpha(t)$ is an increasing deterioration function with $\alpha(0) \geq 0$. He proved that the single-machine problems to minimize makespan and the sum of squared completion times are polynomially solvable. In addition, he showed that the problems to minimize the weighted sum of completion times and maximum lateness can be solved by the weighted shortest processing time (WSPT) rule and the earliest due date (EDD) rule for the case that all the jobs have a common basic processing time or the case that the basic processing times and the weights (or due dates) are agreeable. Furthermore, Wang and Cheng (2007a) studied the machine scheduling problems with the effects of deterioration and learning. In this model the processing times of jobs are defined as functions of their starting times and positions in a sequence, i.e., $p_{jr} = (p_j + \alpha_j t)r^a$, where p_j is the basic processing time and α_j is the deterioration rate of job *j*. They introduced polynomial-time solutions for some single-machine problems and flowshop problems. Wang and Cheng (2007b) considered a model in which the actual processing time is $(p_0 + \alpha_j t)r^a$, where p_0 is a common basic processing time, α_j is the growth rate, *r* is the scheduled position, and *a* is the learning index. They studied the single-machine problem to minimize makespan and showed that the schedule produced by the largest growth rate rule is unbounded for their model, although it is optimal for the scheduling problem with deteriorating jobs and no learning.

In this paper we study a new scheduling model with deteriorating jobs and learning effects. Under the proposed model, the actual processing time of a job depends not only on the total normal processing times of the jobs already processed, but also on its scheduled position. The remainder of this paper is organized as follows. We present in the next section the solution procedures for the single-machine problems to minimize makespan, total completion time, total weighted completion time, and maximum lateness. In Section 3 we consider some special cases of the problems to minimize makespan and total completion time in the permutation flowshop environment. We conclude the paper in the last section.

2. Some single-machine problems

A practical example that motivates the above scheduling model is the manual production of glass crafts by a skilled craftsman. Silicon-based raw material is first heated up in an oven until it becomes a lump of malleable dough from which the craftsman cuts pieces and shapes them according to different designs into different glass craft products. The initial time to heat up the raw material to the threshold temperature at which it can be shaped is long and so the first piece (i.e., job) has a long processing time, which includes both the heating time (i.e., the deterioration effect) and the shaping time (i.e., the normal processing time). The second piece requires a shorter time to re-heat the dough to the threshold temperature (i.e., a smaller deterioration effect). Similarly, the later a piece is cut from the dough, the shorter is its heating time to reach the threshold temperature. On the other hand, the pieces that are shaped later require shorter shaping times because the craftsman's productivity improves as a result of learning.

Formulation of the scheduling model with deteriorating jobs and learning effects in the single-machine case is as follows. There are n simultaneously ready jobs to be processed on a single machine. Each job i has a normal processing time p_i and a due date d_i. Due to the learning and deteriorating effects, the actual processing time of job j is modelled as

$$p_{j[r]} = p_j \left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2}, \tag{1}$$

if it is scheduled in the *r*th position in a sequence, where $p_{[l]}$ denotes the normal processing time of the job scheduled in the *l*th position in the sequence, $p_0 > 0$ is a given parameter, and a_1 and a_2 denote the deteriorating and learning indices with $a_1 < 0$ and $a_2 < 0$. In this model, the actual processing time of a job becomes shorter if it is scheduled in a later position as a result of learning. On the other hand, due to the effect of deterioration, the actual processing time of a job becomes longer while awaiting processing. However, the rate of deterioration decreases with the waiting time.

Before presenting the main results, we first present several lemmas that will be used in the proofs of the theorems in the sequel. The proofs of the lemmas are given in the Appendix.

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

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 $a_1 < 0$, $a_2 < 0$, $\lambda \geq 1$, $x \geq 0$, and $r = 1, 2, ..., n-1$.

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

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 $k \geq 1$, $x \geq 0$, $a_1<0$, $a_2<0$, and $r=1, 2, \cdots, n-1$.

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 $k \geq 1, x \geq 0$, $\lambda \geq 1$, $a_1<0$, $a_2<0$, and $r=1, 2, \cdots, n-1$.

We will prove the following theorem using the standard pairwise interchange method. Suppose that S_1 and S_2 are two given job schedules. The difference between S_1 and S_2 is a pairwise interchange of two adjacent jobs *i* and *j*. That is, $S_1=(\sigma, i, j, \sigma')$ and $S_2=(\sigma, j, i, \sigma')$, where σ and σ' each denote a partial sequence. It is said that S_1 dominates S_2 if the objective function under S_1 is less than that under S_2. Furthermore, we assume that there are *r*-1 jobs in σ. Thus, jobs *i* and *j* are the *r*th and (*r*+1)th job in S_1, whereas jobs *j* and *i* are scheduled in the *r*th and (*r*+1)th position in S_2. In addition, let *A* denote the completion time of the last job in σ. Under S_1, the completion times of jobs *i* and *j* are respectively

$$C_i(S_1)=A+p_i\left(\frac{p_0+\sum_{l=1}^{r-1} p_{[l]}}{p_0+\sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2}, \tag{2}$$

and

$$C_j(S_1)=A+p_i\left(\frac{p_0+\sum_{l=1}^{r-1} p_{[l]}}{p_0+\sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2}+p_j\left(\frac{p_0+\sum_{l=1}^{r-1} p_{[l]}+p_i}{p_0+\sum_{l=1}^{n} p_l}\right)^{a_1}(r+1)^{a_2}. \tag{3}$$

Similarly, the completion times of jobs *j* and *i* in S_2 are respectively

$$C_j(S_2) = A + p_j\left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2} \tag{4}$$

and

$$C_i(S_2) = A + p_j\left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2} + p_i\left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]} + p_j}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} (r+1)^{a_2}. \tag{5}$$

Theorem 1. For the $1/p_{j[r]} = p_j\left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2} / C_{\max}$ problem, the optimal schedule is obtained by sequencing the jobs in the shortest processing time (SPT) order.

Proof. Suppose that $p_i \le p_j$. To show that S_1 dominates S_2, it suffices to show that $C_j(S_1) \le C_i(S_2)$.

Taking the difference between Equations (3) and (5), we have

$$\begin{aligned} C_i(S_2) - C_j(S_1) = {} & (p_j - p_i)\left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2} + p_i\left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]} + p_j}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} (r+1)^{a_2} \\ & - p_j\left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]} + p_i}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} (r+1)^{a_2}. \end{aligned} \tag{6}$$

Substituting $t = \frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}$, $w = \frac{p_i}{p_0 + \sum_{l=1}^{n} p_l}$, $\lambda = \frac{p_j}{p_i}$, and $x = \frac{w}{t}$ into Equation

(6), we have

$$C_i(S_2) - C_j(S_1) = p_i t^{a_1} r^{a_2} \{\lambda[1-(1+x)^{a_1}(\frac{r+1}{r})^{a_2}] - [1-(1+\lambda x)^{a_1}(\frac{r+1}{r})^{a_2}]\}. \quad (7)$$

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

$$C_i(S_2) - C_j(S_1) \geq 0. \quad (8)$$

Thus, S_1 dominates S_2. 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 $1/p_{j[r]} = p_j(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l})^{a_1} r^{a_2} / \sum_{l=1}^{n} C_l$ problem, the optimal schedule is obtained by sequencing jobs in the SPT order.

Proof. The proof is similar to that of Theorem 1 and is omitted.

Smith (1956) showed that sequencing jobs according to the WSPT rule provides an optimal schedule for the classical single-machine scheduling problem to minimize total weighted completion time, i.e., sequencing jobs in non-decreasing order of p_j / w_j, where w_j is the weight of job *j*. The following theorem shows that the WSPT order remains optimal for our scheduling model with deteriorating jobs and learning effects if the processing times and the weights are agreeable, i.e., $\frac{p_j}{p_i} \geq \frac{w_j}{w_i} \geq 1$ for all jobs *i* and *j*.

Theorem 3. For the $1/p_{j[r]} = p_j(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}/\sum_{l=1}^{n}w_lC_l$ problem, an optimal schedule is obtained by sequencing jobs in non-decreasing order of p_i/w_i (i.e., in the WSPT order) if the processing times and the weights are agreeable, i.e., $\frac{p_j}{p_i}\geq\frac{w_j}{w_i}\geq 1$ for all jobs *i* and *j*.

Proof. Suppose that $\frac{p_j}{p_i}\geq\frac{w_j}{w_i}\geq 1$. It is seen from Theorem 1 that $C_j(S_1)<C_i(S_2)$ since $p_i\leq p_j$. Thus, to show that S_1 dominates S_2, it suffices to show that $w_iC_i(S_1)+w_jC_j(S_1)\leq w_jC_j(S_2)+w_iC_i(S_2)$. From Equations (2) to (5), we have

$$[w_jC_j(S_2)+w_iC_i(S_2)]-[w_iC_i(S_1)+w_jC_j(S_1)]$$

$$=\{w_j[A+p_j(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}+w_i[A+p_j(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}$$

$$+p_i(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}+p_j}{p_0+\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}]\}-\{w_i[A+p_i(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}$$

$$+w_j[A+p_i(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}+p_j(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}+p_i}{p_0+\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}]\}$$

$$=(w_ip_j-w_jp_i)(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}+w_jp_j[(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}$$

$$-(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}+p_i}{p_0+\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}]-w_ip_i[(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}$$

$$-(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}+p_j}{p_0+\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}]. \tag{9}$$

Substituting $\lambda=\frac{p_j/w_j}{p_i/w_i}\geq 1$, $t=\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l}$, $x=\frac{p_i}{p_0+\sum_{l=1}^{r-1}p_{[l]}}$, and $k=\frac{w_j}{w_i}$ into Equation (9), we have

$$[w_jC_j(S_2)+w_iC_i(S_2)]-[w_iC_i(S_1)+w_jC_j(S_1)]$$

$$=w_jp_it^{a_1}r^{a_2}\{(\lambda-1)+\lambda k[1-(1+x)^{a_1}(\frac{r+1}{r})^{a_2}]-\frac{1}{k}[1-(1+\lambda kx)^{a_1}(\frac{r+1}{r})^{a_2}]\} \tag{10}$$

From Lemma 5, we have $[w_jC_j(S_2)+w_iC_i(S_2)]\geq[w_iC_i(S_1)+w_jC_j(S_1)]$.

Thus, repeating this interchange argument for all the jobs not sequenced in the WSPT order completes the proof of Theorem 3.

Sequencing jobs according to the EDD rule provides an optimal sequence for the classical single-machine scheduling problem to minimize maximum lateness. We show in the following that under the proposed model, the EDD order provides an optimal solution for the problem to minimize maximum lateness 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.

Theorem 4. For the $1/p_{j[r]}=p_j(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}/L_{\max}$ problem, an optimal schedule is obtained by sequencing jobs in non-decreasing order of d_i (i.e., in 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 $C_j(S_1) < C_i(S_2)$. To show that S_1 dominates S_2, it suffices to show that $\max\{L_i(S_1), L_j(S_1)\} \le \max\{L_i(S_2), L_j(S_2)\}$, where $L_i(S_k)$ and $L_j(S_k)$ denote the lateness of jobs *i* and *j* under schedule S_k for *k*= 1, 2, respectively. By definition, the lateness of jobs *i* and *j* in S_1 and jobs *j* and *i* in S_2 are respectively

$$L_i(S_1) = C_i(S_1) - d_i, \tag{11}$$

$$L_j(S_1) = C_j(S_1) - d_j, \tag{12}$$

$$L_j(S_2) = C_j(S_2) - d_j, \tag{13}$$

and

$$L_i(S_2) = C_i(S_2) - d_i. \tag{14}$$

Since $p_i \le p_j$, we have from Theorem 1 that

$$C_j(S_1) < C_i(S_2). \tag{15}$$

From $d_i \le d_j$, we have

$$L_j(S_1) \le L_i(S_2). \tag{16}$$

Since job *i* is processed before job *j* in *S*, we have from Equation (15) that

$$L_i(S_1) \le L_i(S_2). \tag{17}$$

From Equations (16) and (17), we have

$$\max\{L_i(S_1), L_j(S_1)\} \le \max\{L_i(S_2), L_j(S_2)\}.$$

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

3. Flowshop problems

Formulation of the scheduling model with deteriorating jobs and learning effects for the case of a flowshop is as follows. Suppose that there is a set of n jobs to be processed on m machines M_1, M_2, …, M_m. Each job j consists of m operations O_{1j}, O_{2j}, …, O_{mj}, where O_{ij} has to be processed on machine M_i, i = 1, 2, …, m. The 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. The normal processing time of O_{ij} is denoted by p_{ij}. The actual processing time of job j on machine M_i if it is scheduled in the rth position in a sequence is

$$p_{ij[r]} = p_{ij}\left(\frac{p_{i0} + \sum_{l=1}^{r-1} p_{i[l]}}{p_{i0} + \sum_{l=1}^{n} p_{il}}\right)^{a_1} r^{a_2}, \tag{18}$$

where p_{i0} is a given parameter, a_1 and a_2 denote the deteriorating and the learning indices with $a_1 < 0$ and $a_2 < 0$. For a given schedule π, let $C_{ij} = C_{ij}(\pi)$ denote the completion time of operation O_{ij}, and $C_j = C_{mj}$ denote the completion time of job j. For the traditional m-machine permutation flowshop problem, Pinedo

(2002) showed that if the normal processing times of any job on all the machines are identical, i.e., $p_{ij} = p_j$, then the completion time of the *j*th job in a given sequence *S* is as follows:

$$C_{[j]}(S) = \sum_{k=1}^{j} p_{[k]} + (m-1)\max\{p_{[1]}, p_{[2]}, ..., p_{[j]}\}. \tag{19}$$

Similarly, in the *m*-machine permutation flowshop environment under the proposed model, if the normal processing times of any job on all the machines are identical, i.e., $p_{ij} = p_j$, then we can derive that the completion time of the *j*th job in a given sequence *S* is

$$C_{[j]}(S) = \sum_{k=1}^{j} \left(\frac{p_0 + \sum_{l=1}^{k-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} k^{a_2} p_{[k]}$$

$$+(m-1)\max\{p_{[1]}, \left(\frac{p_0 + p_{[1]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} 2^{a_2} p_{[2]}, ..., \left(\frac{p_0 + \sum_{l=1}^{j-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} j^{a_2} p_{[j]}\}. \tag{20}$$

Theorem 5. For the $Fm / p_{ij[r]} = p_j \left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2} / C_{\max}$ problem, an optimal schedule is obtained by sequencing jobs in the SPT order.

Proof. Suppose that S_1 and S_2 are two job schedules. The difference between S_1 and S_2 is a pairwise interchange of two adjacent jobs *i* and *j*, i.e., $S_1 = (\sigma, i, j, \sigma')$ and $S_2 = (\sigma, j, i, \sigma')$, where σ and σ' each denote a partial sequence. Furthermore, we assume that there are *r*-1 jobs in σ. Thus, jobs *i* and *j* are the *r*th and (r+1)th job in S_1, whereas jobs j and i are scheduled in the rth and (r+1)th position in S_2. In addition, let A denote the completion time of the last job in σ. Under S_1, the completion time of job j is

$$C_j(S_1) = A + p_i\left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2} + p_j\left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]} + p_i}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} (r+1)^{a_2}$$

$$+(m-1)\max\{p_{[1]},\ldots,\left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2} p_i, \left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]} + p_i}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} (r+1)^{a_2} p_j\}. \quad (21)$$

Similarly, the completion time of job i in S_2 is

$$C_i(S_2) = A + p_j\left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2} + p_i\left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]} + p_j}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} (r+1)^{a_2}$$

$$+(m-1)\max\{p_{[1]},\ldots,\left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2} p_j, \left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]} + p_j}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} (r+1)^{a_2} p_i\}. \quad (22)$$

Suppose that $p_i \le p_j$. To show that S_1 dominates S_2, it suffices to show that $C_j(S_1) \le C_i(S_2)$. Since $p_i \le p_j$, $a_1 < 0$, and $a_2 < 0$, we have

$$\left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2} p_j \ge \left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2} p_i, \quad (23)$$

and

$$\left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2} p_j \ge \left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]} + p_i}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} (r+1)^{a_2} p_j. \quad (24)$$

This implies that

$$\max\{p_{[1]},\ldots,(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}p_j,(\frac{p_0+\sum_{l=1}^{j-1}p_{[l]}+p_j}{p_0+\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}p_i\}$$

$$\geq\max\{p_{[1]},\ldots,(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}p_i,(\frac{p_0+\sum_{l=1}^{j-1}p_{[l]}+p_i}{p_0+\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}p_j\}. \tag{25}$$

From Equations (21) and (22), we have

$$\begin{aligned}C_i(S_2)-C_j(S_1)=\{&A+p_j(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}+p_i(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}+p_j}{p_0+\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}\\
&+(m-1)\max\{p_{[1]},\ldots,(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}p_j,(\frac{p_0+\sum_{l=1}^{j-1}p_{[l]}+p_j}{p_0+\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}p_i\}\\
&-\{A+p_i(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}+p_j(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}+p_i}{p_0+\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}\\
&+(m-1)\max\{p_{[1]},\ldots,(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}p_i,(\frac{p_0+\sum_{l=1}^{j-1}p_{[l]}+p_i}{p_0+\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}p_j\}.\end{aligned} \tag{26}$$

Substituting $t=\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l}$, $\lambda=\frac{p_j}{p_i}$, $w=\frac{p_i}{p_0+\sum_{l=1}^{n}p_l}$, and $x=\frac{w}{t}$ into Equation (26), we have

$$C_i(S_2)-C_j(S_1)=t^{a_1}r^{a_2}p_i\{\lambda[1-(1+x)^{a_1}(\frac{r+1}{r})^{a_2}]-[1-(1+\lambda x)^{a_1}(\frac{r+1}{r})^{a_2}\}$$

$$+(m-1)(\max\{p_{[1]},\ldots,(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}p_j,(\frac{p_0+\sum_{l=1}^{j-1}p_{[l]}+p_j}{p_0+\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}p_i\}$$

$$-\max\{p_{[1]},\ldots,(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}p_i,(\frac{p_0+\sum_{l=1}^{j-1}p_{[l]}+p_i}{p_0+\sum_{l=1}^{n}p_l})^{a_1}(r+1)^{a_2}p_j\}). \qquad (27)$$

From Equation (25) and Lemmas 1 and 2, we have

$$C_i(S_2)-C_j(S_1)\geq 0. \qquad (28)$$

Thus, S_1 dominates S_2. 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 $Fm/p_{ij[r]}=p_j(\frac{p_0+\sum_{l=1}^{r-1}p_{[l]}}{p_0+\sum_{l=1}^{n}p_l})^{a_1}r^{a_2}/\sum_{l=1}^{n}C_l$ problem, an optimal schedule is obtained by sequencing jobs in the SPT order.

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

4. Conclusions

The main contribution of this paper is to provide the optimal solutions for several scheduling problems where the phenomena of job deterioration and learning exist simultaneously. We showed that the single-machine problems are polynomially solvable if the performance criterion is makespan, total completion time, total weighted completion time, or maximum lateness. Moreover, we showed that the flowshop permutation problems are polynomially solvable under a certain condition. Further research may focus on other performance criteria or extension of the problems under study in this paper to other shop problems.

Acknowledgements

We are grateful to two anonymous referees for their constructive comments on an earlier version of our paper.

References

Alidaee, B., & Womer, N.K. (1999). Scheduling with time dependent processing times: Review and extensions. Journal of the Operational Research Society, 50, 711-720.

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

Biskup, D. (2007). A state-of-the-art review on scheduling with learning effect. European Journal of Operational Research (in press).

Browne, S., & Yechiali, U. (1990). Scheduling deteriorating jobs on a single processor. Operations Research, 38, 495-498.

Cheng, T.C.E., Ding, Q., & Lin, B.M.T. (2004). A concise survey of scheduling with time-dependent processing times. European Journal of Operational Research, 152, 1-13.

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

Gupta, J.N.D., & Gupta, S.K. (1988). Single facility scheduling with nonlinear processing times. Computers and Industrial Engineering, 14, 387-393.

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

Kunnathur, A.S., & Gupta, S.K. (1990). Minimizing the makespan with late start penalties added to processing times in a single facility scheduling problem. European Journal of Operational Research, 47, 56-64.

Lee, W.C. (2004). A note on deteriorating jobs and learning in single-machine scheduling problems. International Journal of Business and Economics, 3, 83-89.

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

Russell, R., & Taylor, III.B.W. (2000). Operations Management: Multimedia Version, 3rd edition. Prentice Hall: Upper Saddle River, NJ.

Smith, W.E. (1956). Various optimizers for single state production. Naval Research

Logistic Quarterly, 3, 59-66.

Wang, J.B. (2006). A note on scheduling problems with learning effect and deteriorating jobs. International Journal of Systems Science, 37, 827-833.

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

Wang, J.B., & Cheng, T.C.E. (2007a). Scheduling problems with the effects of deterioration and learning. Asia-Pacific Journal of Operational Research, 24, 1-17.

Wang, X., & Cheng, T.C.E. (2007b). Single-machine scheduling with deteriorating jobs and learning effects to minimize the makespan. European Journal of Operational Research, 178, 57-70.

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

Yelle, L.E. (1979). The learning curve: historical review and comprehensive survey. Decision Science, 10, 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 $a_1<0$, $a_2<0$, $x\geq 0$ 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 $x\geq 0$, $a_1<0$, $a_2<0$, and $r=1,2,...,n-1$. Thus, this implies that $f(x)$ is a non-decreasing function on $x\geq 0$. 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 $x\geq 0$, $a_1<0$, $a_2<0$, 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 $a_1<0$, $a_2<0$, $\lambda\geq 1$, $x\geq 0$, 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 $a_1 < 0$, it implies that $g''(\lambda) \geq 0$. Therefore, $g'(\lambda)$ is a non-decreasing function for $\lambda \geq 1$. From Lemma 1, we have

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

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

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

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

$$g(\lambda) \geq 0$$

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

Lemma 3. $1+k[1-(1+x)^{a_1}(\frac{r+1}{r})^{a_2}] + a_1 x(1+kx)^{a_1-1}(\frac{r+1}{r})^{a_2} \geq 0$ for $a_1 < 0$, $a_2 < 0$, $k \geq 1$, $x \geq 0$, 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

$$\begin{aligned} f'(x) &= -ka_1(1+x)^{a_1-1}(\frac{r+1}{r})^{a_2} + a_1(1+kx)^{a_1-1}(\frac{r+1}{r})^{a_2} + a_1(a_1-1)kx(1+kx)^{a_1-2}(\frac{r+1}{r})^{a_2} \\ &= a_1[-k(1+x)^{a_1-1} + (1+a_1kx)(1+kx)^{a_1-2}](\frac{r+1}{r})^{a_2}. \end{aligned}$$

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

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

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

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

The proof is completed.