Single-machine Scheduling with Deteriorating Jobs and Learning Effects to Minimize the Makespan

Xiuli Wang T. C. Edwin Cheng [#]

Department of Logistics
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
E-mail: LGTcheng@polyu.edu.hk

Abstract

This paper studies the single-machine scheduling problem with deteriorating jobs and learning considerations. The objective is to minimize the makespan. We first show that the schedule produced by the largest growth rate rule is unbounded for our model, although it is an optimal solution for the scheduling problem with deteriorating jobs and no learning. We then consider three special cases of the problem, each corresponding to a specific practical scheduling scenario. Based on the derived optimal properties, we develop an optimal algorithm for each of these cases. Finally, we consider a relaxed model of the second special case, and present a heuristic and analyze its worst-case performance bound.

Keywords: Scheduling, single-machine, deteriorating jobs, learning effects, makespan.

[#]Corresponding author.

1. Introduction

Machine scheduling problems with deteriorating jobs have received increasing attention in resent years. The actual processing time of a job in a schedule is modeled as an increasing function of its starting time due to deterioration effects. This model reflects a variety of real-life situations such as steel production, resource allocation, fire fighting, maintenance or cleaning, etc. (see Kunnathur and Gupta 1990, Mosheiov 1995), in which any delay in processing a job may result in an increasing effort to accomplish the job. In order to make the analysis possible, most research models the actual processing time of a job as a linear or piecewise linear increasing function of its starting time. For single-machine scheduling problems, Browne and Yechiali (1990) assumed that the actual processing time is $p_i + \alpha_i t$, where p_i is the basic processing time, α_i is the growth rate of the processing time, and *t* is the starting time, of job *i*. They showed that sequencing the jobs in increasing order of p_i / α_i minimizes the makespan. Mosheiov (1991) considered the total flow time minimization problem with the actual processing time of job *i* being $p_0 + \alpha_i t$, where p_0 is a common basic processing time, and α_i is the growth rate of the processing time, of job *i*. He showed that an optimal schedule is V-shaped with respect to α_i, i.e., the jobs appearing before the minimal growth rate job are sequenced in nonincreasing order of α_i and the ones after it are sequenced in nondecreasing order of α_i. Mosheiov (1994) further simplified the model with the actual processing time of job *i* being $\alpha_i t$, and showed that the problems of minimizing such objectives as the makespan, total flow time, sum of weighted completion times, total lateness, number of tardy jobs, maximum lateness, and maximum tardiness are all polynomially solvable. Bachman and Janiak (2000) proved that the maximum lateness minimization problem with the actual processing time of job *i* being $p_i + \alpha_i t$ is NP-hard, and presented two heuristics for this problem. For this model, Bachman et al. (2002) proved that minimizing the total weighted completion time is NP-hard. For research results on other scheduling models considering deterioration effects and under different machine environments, the reader may refer to the review papers of Alidaee and Womer (1999), and Cheng et al. (2004).

On the other hand, it is reasonable and necessary to consider learning effects in scheduling research under some practical situations. Biskup (1999), and Cheng and Wang (2000) have observed and analyzed various production activities for which scheduling with learning effects may arise. Cheng and Kovalyov (1994) were probably the first researchers who introduced the concept of learning into scheduling. They modeled the learning effects on the actual processing time of a job as a piecewise linear decreasing function of the total actual processing time of all the jobs scheduled before it. They studied both single-machine and parallel-machine problems to minimize the makespan and total flow time. Cheng and Wang (2000) used the volume-dependent processing time function to model the learning effects, in which the learning effects on the processing time of a job depend on the number of jobs processed before the job. They showed that the maximum lateness minimization problem on a single-machine is NP-hard in the strong sense, and developed two bounded heuristics. Different from the above models to deal with the learning effects, Biskup (1999) used the log-linear learning curve popularized in industrial engineering to describe the learning effects, i.e., if job i is scheduled in position r in a sequence, its actual processing time is $p_i r^a$, where p_i is the basic processing time and $a \leq 0$ is the learning index (to be defined later). He showed that single-machine scheduling problems to minimize the total flow time and total deviations from a common due date are polynomially solvable. Mosheiov (2001) followed Biskup's (1999) model and showed that the single-machine makespan minimization problem remains polynomially solvable. He also provided some counterexamples to show that the optimal properties for the corresponding counterpart classical scheduling problems no longer hold. For instance, the earliest due date rule is not optimal for minimizing the maximum lateness, and the weighted shortest processing time rule is not optimal for minimizing the weighted flow time.

However, to the best of our knowledge, there exist only a few research results on scheduling models considering the effects of deterioration and learning at the same time, although the phenomena can be found in many real-life situations. Wang and Cheng (2005) discussed several real-life examples of a processing environment involving task rotation, where job deterioration is caused by the forgetting effects, while the learning effects reflect that the workers become more skilled to operate the machines through experience accumulation. Here, we give another practical example. The main stage in the production of porcelain craftworks is to shape the raw material according to the 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 design and operations. For this situation, considering both the job deterioration and learning effects in job scheduling is both necessary and reasonable. In this paper we study a single-machine scheduling problem considering both deterioration and learning effects to minimize the makespan.

In Section 2 we will give a formal description of the model under study. In Section 3 we will show that the schedule generated by the largest growth rate rule is unbounded for our model. We will identify three special cases of the problem, each corresponding to a specific practical scheduling scenario, and develop an optimal algorithm for each case in Section 4. In Section 5 we will consider a relaxed model of one of the cases in Section 4, and present a heuristic and analyze its worst-case performance bound. Finally, conclusions are given in Section 6.

2. Formulation

To model the effect of job deterioration, we follow Mosheiov (1991) by assuming that the processing time of a job is a linear function of its starting time. The learning effect is modeled in its popular form of the log-linear curve (see, for example, Biskup, 1999). In order to study the effects of deterioration and learning simultaneously, we combine the above models to constitute our model. Formally, the model is stated as follows:

There are n jobs to be scheduled on a single machine. All the jobs are nonpreemptive and available for processing at time zero. The machine can handle at most one job at a time and cannot stand idle until the last job assigned to it has finished processing. If job $i, i = 1, 2, \cdots, n,$ is scheduled in position r in a sequence, its actual processing time is

$$p_{i,r} = (p_0 + \alpha_i t) r^a, \tag{1}$$

where p_0 is a common basic processing time that is incurred if job i is scheduled first in a sequence; t is the starting time of job i to be processed; α_i is the growth rate of the processing time of job i, which is the amount of increase in the processing time of job i per unit delay in its starting time due to the deterioration effects; and a is the learning index, given as the logarithm to the base 2 of the learning rate x, i.e., $a = \log_2 x, x \in (0, 100\%).$ The assumption that all the jobs have an identical basic processing time is reasonable, which both reflects some real-life situations and may serve as an approximation of the more general cases. In addition, this assumption is necessary to make the analysis possible from a modeling perspective.

Our objective is to schedule the jobs so as to minimize the makespan, i.e., the completion time of the last job. Noting that an optimal schedule not only depends on the performance measure, but also on the parameter distribution of the growth rates and the learning index, we give an exact description of the parameter distribution of our model as follows:

$$GL = \{\{\alpha_1, \alpha_2, \cdots, \alpha_n\}, \{a\} \,|\, \alpha_1 > 0, \alpha_2 > 0, \cdots, \alpha_n > 0, a < 0\}.$$

Adopting the three-field notation of Graham et al. (1979) to describe classical scheduling problems, we denote our problem as $1 \,|\, p_{i,r} = (p_0 + \alpha_i t) r^a, GL \,|\, C_{\max}$.

For a given schedule $\pi = [1, 2, \cdots, n]$, we can derive an expression of its objective function. Setting $p_0 = 1$, from (1), we have

$$p_{1,1} = (1 + \alpha_1 \cdot 0) 1^a = 1^a,$$

$$p_{2,2} = (1+\alpha_2 p_{1,1})2^a = 2^a + \alpha_2 2^a p_{1,1},$$

then,

$$p_{1,1} + p_{2,2} = p_{1,1} + (2^a + \alpha_2 2^a p_{1,1}) = 2^a + 1^a(1+\alpha_2 2^a).$$

Furthermore,

$$p_{3,3} = [1+\alpha_3(p_{1,1}+p_{2,2})]3^a = 3^a + \alpha_3 3^a (p_{1,1}+p_{2,2}),$$

then,

$$\begin{aligned} p_{1,1} + p_{2,2} + p_{3,3} &= (p_{1,1}+p_{2,2}) + [3^a + \alpha_3 3^a (p_{1,1}+p_{2,2})] \\ &= 3^a + (p_{1,1}+p_{2,2})(1+\alpha_3 3^a) \\ &= 3^a + 2^a(1+\alpha_3 3^a) + 1^a(1+\alpha_2 2^a)(1+\alpha_3 3^a). \end{aligned}$$

Generally, we have

$$\sum_{i=1}^{n} p_{i,i} = \sum_{k=1}^{n}[k^a \prod_{i=k+1}^{n}(1+\alpha_i i^a)],$$

where $\prod_{i=n+1}^{n}(1+\alpha_i i^a) := 1$.

If the common basic processing time is p_0, then the makespan $C_{\max}(\pi)$ can be expressed as

$$C_{\max}(\pi) = p_0 \sum_{k=1}^{n}[k^a \prod_{i=k+1}^{n}(1+\alpha_i i^a)]. \tag{2}$$

Analyzing (1), we see that the actual processing time of a job is not only related to its position in a schedule, but also to the jobs processed prior to it. Hence, the optimal properties with respect to the learning effects only are not applicable to our problem. On the other hand, the actual processing time of a job is no longer a linear function of its starting time, so the existing results for the optimal schedule concerning only the linear deterioration of jobs no longer hold for our scheduling model either. These characteristics make our model very difficult to deal with.

We also observe from (1) that although a job has a distinct actual processing time if it is sequenced in a different position in a schedule, the jobs can be distinguished from one another by their growth rates; in other words, we can identify a job by its growth rate. Based on this observation, we may say that the expression (2) for the makespan is appropriate for any schedule in which the jobs are numbered by their positions in the schedule.

3. The general model

To the best of our knowledge, the complexity of the model under study is open. The largest growth rate (LGR) rule, which sequences the jobs in nonincreasing order of α_i, yields an optimal solution for $1 | p_i = p_0 + \alpha_i t | C_{\max}$ (Browne and Yechiali, 1990). However, the LGR rule is unbounded for our model.

We define a subset of *GL* as

$$\Pi = \{\{\alpha_1, \alpha_2, \cdots, \alpha_n\}, \{a\} \mid \alpha_1 > (n+1)^2 / n,\ \alpha_k = (n-k+2)^2 / n$$

$$\text{for } k = 2, \cdots, n,\ a = -2\}.$$

For a job system consisting of n jobs with parameters belonging to Π, let $\pi = [1, 2, \cdots, n]$ be a schedule produced by the LGR rule. Then we have

$$C_{\max}(\pi) = p_0\{1^{-2}(1 + \frac{n^2}{n} \cdot 2^{-2})(1 + \frac{(n-1)^2}{n} \cdot 3^{-2}) \cdots (1 + \frac{2^2}{n} \cdot n^{-2})$$

$$+ 2^{-2}(1 + \frac{(n-1)^2}{n} \cdot 3^{-2})(1 + \frac{(n-2)^2}{n} \cdot 4^{-2}) \cdots (1 + \frac{2^2}{n} \cdot n^{-2})$$

$$+ \cdots + (n-1)^{-2}(1 + \frac{2^2}{n} \cdot n^{-2}) + n^{-2}\}$$

$$> p_0(1 + \frac{n}{4}).$$

We consider a schedule $\pi' = [1, n, n-1, \cdots, 2]$, whose makespan is

$$C_{\max}(\pi') = p_0\{1^{-2}(1 + \frac{2^2}{n} \cdot 2^{-2})(1 + \frac{3^2}{n} \cdot 3^{-2}) \cdots (1 + \frac{n^2}{n} \cdot n^{-2})$$

$$+ 2^{-2}(1 + \frac{3^2}{n} \cdot 3^{-2})(1 + \frac{4^2}{n} \cdot 4^{-2}) \cdots (1 + \frac{n^2}{n} \cdot n^{-2})$$

$$+\cdots+(n-1)^{-2}(1+\frac{n^2}{n}\cdot n^{-2})+n^{-2}\}. \qquad (3)$$

Since $\left(1+\frac{1}{n}\right)^{n-1} < e$, where e is the base of the natural logarithm, from (3), we have

$$C_{\max}(\pi') < p_0 e(1+2^{-2}+3^{-2}+\cdots+n^{-2}) < 2p_0 e.$$

Since the optimal objective $C^*_{\max} \le C_{\max}(\pi')$, then

$$\frac{C_{\max}(\pi)}{C^*_{\max}} \ge \frac{C_{\max}(\pi)}{C_{\max}(\pi')} > \frac{p_0(1+1/4)}{2p_0 e} = \frac{1}{2e}(1+\frac{n}{4}).$$

For any given number $M > 0$, if $n > 4(2eM-1)$, we have

$$\frac{C_{\max}(\pi)}{C^*_{\max}} > M.$$

Summarizing the above analysis, for any given number $M > 0$, there exists a job system with parameters belonging to Π such that the makespan of the LGR rule schedule is larger than M times of that of the optimal schedule. So we have the following theorem.

Theorem 1. The LGR rule schedule is unbounded for the problem $1 \mid p_{i,r} = (p_0+\alpha_i t)r^a,\ GL \mid C_{\max}$.

Theorem 1 indicates that the LGR rule is no longer effective in solving the general problem under study.

4. Some polynomially solvable cases

To further observe our model, we find that the model processes a very intricate geometric structure due to the fact that learning and deterioration create opposing effects on the objective function qualitatively and quantitatively. Hence, the analysis will be facilitated by classifying the general problem into some special cases according to different distributions of the growth rates and the learning index.

In the following we define three special cases characterized by the distributions of the parameters $\alpha_1, \alpha_2, \cdots, \alpha_n$ and *a*.

First, we define *GL*1 as a subset of *GL* as follows:

$$GL1 = \{\{\alpha_1, \alpha_2, \cdots, \alpha_n\}, \{a\} \mid \alpha_1 > 0, \alpha_2 >, \cdots, \alpha_n > 0, a \le -1\}.$$

This case is denoted as $1 \mid p_{i,r} = (p_0 + \alpha_i t) r^a, GL1 \mid C_{\max}$.

From $a \le -1$ and $a = \log_2 x$, the learning rate *x* may be a percentage between 0 and 50%. This case describes the situation in which the operator has a good learning ability and learning has a more significant effect on shortening the processing times of the jobs than other factors.

Second, we impose on the growth rates of the jobs in a job system and the learning index the following constraints:

$$\begin{cases} \alpha_1 \ge \max\{\alpha_2, \alpha_3, \cdots, \alpha_n\}, \\ \max\{\alpha_2, \alpha_3, \cdots, \alpha_n\} \le \dfrac{2^{-a}}{n}, \\ -\dfrac{1}{en} \le a < 0, \text{ where } e \text{ is the base of the natural logrithm}. \end{cases}$$

The parameter distribution subset of *GL* satisfying the above constraints is denoted as *GL*2. The schedule problem in this situation is denoted as $1 \mid p_{i,r} = (p_0 + \alpha_i t) r^a$, $GL2 \mid C_{\max}$.

In this case, the growth rates of all the jobs are small and the learning rate is close to 100%. This model reflects the practical situation in which the deteriorating effects on all the jobs are not very significant and the growth rate differences among the jobs are not large. The learning efforts to shorten the actual processing times of the jobs are very difficult.

Third, for the parameter distributions of the growth rates of all the jobs and the learning index, we define a subset *GL*3 of *GL* as

$$\begin{cases} \alpha_1 \ge \max\{\alpha_2, \alpha_3, \cdots, \alpha_n\}, \\ \alpha_3 \ge 3^{-a}\beta, \alpha_{i+1} - \alpha_i \ge [(i+1)^{-a} - i^{-a}]\beta, \text{where } \beta > 0, i = 3, 4, \cdots, n-1, \\ \alpha_2 \ge \alpha_i \quad \text{for } i^{-a} \ge 2^{-a} + 1 + \dfrac{1}{\beta}, \\ -1 < a < -\dfrac{4}{4+\beta}. \end{cases}$$

The general model with respect to $GL3$ is denoted as $1 | p_{i,r} = (p_0 + \alpha_i t) r^a$, $GL3 | C_{\max}$.

In this case, we clarify some relations between the job growth rates and the learning index by utilizing a parameter β. Once β is determined, $GL3$ stipulates the constraints imposed on the job growth rates and learning index, respectively. In general, this model reflects some real production environments in which the deterioration effects on all the jobs have large differences, and the learning effects on shortening the actual processing times of the jobs are moderate.

In the above definitions, the condition $\alpha_1 \ge \max\{\alpha_2, \alpha_3, \cdots, \alpha_n\}$ in $GL2$ and $GL3$ always holds if the job with the largest growth rate is numbered as job 1. It is noted that $GL1$, $GL2$ and $GL3$ cover all the possible distributions of the learning index a.

For these three cases, we will derive some optimal properties and develop an optimal algorithm for each of them.

4.1 Problem $1 | p_{i,r} = (p_0 + \alpha_i t) r^a, GL1 | C_{\max}$

We notice from (2) that the makespan of the schedule π does not include the growth rate α_1. This means that the contribution of the first job in a schedule to the makespan is a constant, no matter what its growth rate is. Thus, we have the following lemma immediately.

Lemma 1. For the problem $1 | p_{i,r} = (p_0 + \alpha_i t) r^a, GL | C_{\max}$, the job with the largest growth rate should be sequenced in the first position in an optimal schedule.

Lemma 1 tells us which job must be sequenced in the first position in an optimal schedule. In the following we will derive an optimal property for sequencing the remaining $n-1$ jobs.

Lemma 2. For the problem $1 \mid p_{i,r}=(p_0+\alpha_i t)r^a, GL1 \mid C_{\max}$, sequencing the jobs in the 2nd to *n*th positions in nondecreasing order of their growth rates minimizes the makespan.

Proof. For a schedule $\pi = [1, 2, \cdots, n]$, only swap the jobs in positions i ($i>1$) and $i+1$ to generate a new schedule π'. From (2), we have

$$C_{\max}(\pi')-C_{\max}(\pi) = p_0(\alpha_i-\alpha_{i+1})\{[(i+1)^a-i^a][(1+\alpha_2 2^a)(1+\alpha_3 3^a)$$
$$\cdots(1+\alpha_{i-1}(i-1)^a)+2^a(1+\alpha_3 3^a)\cdots(1+\alpha_{i-1}(i-1)^a)+$$
$$\cdots+(i-2)^a(1+\alpha_{i-1}(i-1)^a)+(i-1)^a]+i^a(i+1)^a\}$$
$$(1+\alpha_{i+2}(i+2)^a)\cdots(1+\alpha_n n^a). \quad (4)$$

We set $\min\{\alpha_2 2^a, \alpha_3 3^a, \cdots, \alpha_{i-1}(i-1)^a\}=\varepsilon$. Assuming that $\alpha_i \le \alpha_{i+1}$, from (4), we have

$$C_{\max}(\pi')-C_{\max}(\pi) \ge p_0(\alpha_{i+1}-\alpha_i)\{[i^a-(i+1)^a][(1+\varepsilon)^{i-2}+(1+\varepsilon)^{i-3}2^a$$
$$+\cdots+(1+\varepsilon)(i-2)^a+(i-1)^a]-i^a(i+1)^a\}$$
$$(1+\alpha_{i+2}(i+2)^a)\cdots(1+\alpha_n n^a)$$
$$\ge p_0(\alpha_{i+1}-\alpha_i)\{[i^a-(i+1)^a][(1+\varepsilon)^{i-2}+(1+\varepsilon)^{i-3}$$
$$+\cdots+(1+\varepsilon)+1](i-1)^a-i^a(i+1)^a\}$$
$$(1+\alpha_{i+2}(i+2)^a)\cdots(1+\alpha_n n^a)$$
$$= p_0(\alpha_{i+1}-\alpha_i)\{[i^a-(i+1)^a]\frac{(1+\varepsilon)^{i-1}-1}{\varepsilon}(i-1)^a$$
$$-i^a(i+1)^a\}(1+\alpha_{i+2}(i+2)^a)\cdots(1+\alpha_n n^a)$$

$$\geq p_0(\alpha_{i+1}-\alpha_i)\{[i^a-(i+1)^a]\frac{(1+(i-1)\varepsilon)-1}{\varepsilon}(i-1)^a$$

$$-i^a(i+1)^a\}(1+\alpha_{i+2}(i+2)^a)\cdots(1+\alpha_n n^n)$$

$$= p_0(\alpha_{i+1}-\alpha_i)\{[i^a-(i+1)^a](i-1)^{1+a}$$

$$-i^a(i+1)^a\}(1+\alpha_{i+2}(i+2)^a)\cdots(1+\alpha_n n^a)$$

$$= p_0(\alpha_{i+1}-\alpha_i)i^a(i+1)^a\{[(i+1)^{-a}-i^{-a}](i-1)^{1+a}$$

$$-1\}(1+\alpha_{i+2}(i+2)^a)\cdots(1+\alpha_n n^a). \tag{5}$$

For the function $f(x) = x^{-a}$, its derivative function $\frac{df(x)}{dx}=(-a)x^{-a-1}$. According to the Lagrange finite increment theorem (i.e., if a function $f(x)$ is continuous on the closed interval $[t_1,t_2]$, and differentiable in the open interval (t_1,t_2), then there exists a number $c\in(t_1,t_2)$, such that $f(t_2)-f(t_1)=(t_2-t_1)\frac{df(x)}{dx}|_{x=c}$ holds.), we have

$$(i+1)^{-a}-i^{-a}=(-a)(i+\theta)^{-a-1}, \tag{6}$$

where $0 < \theta < 1$. In $GL1$, $a\leq -1$. Therefore,

$$(i+1)^{-a}-i^{-a}\geq(-a)i^{-a-1}. \tag{7}$$

From (5) and (7), we have

$$C_{\max}(\pi')-C_{\max}(\pi)\geq p_0(\alpha_{i+1}-\alpha_i)i^a(i+1)^a\left\{(-a)\left(\frac{i}{i-1}\right)^{-a-1}-1\right\}$$

$$(1+\alpha_{i+2}(i+2)^a)\cdots(1+\alpha_n n^a),$$

and in $GL1$, $a\leq -1$. So, we have

$$(-a)\left(\frac{i}{i-1}\right)^{-a-1}-1>0.$$

Therefore,

$$C_{\max}(\pi')-C_{\max}(\pi)\geq 0. \quad \square$$

According to Lemmas 1 and 2, we can develop an algorithm for the problem $1|\, p_{i,r} = (p_0+\alpha_i t)r^a,\, GL1\,|C_{\max}$. The algorithm is formally described as follows.

Algorithm AGL1:

Step 1. Select a job with the largest growth rate among the *n* jobs. Sequence it in the first position.

Step 2. For the remaining *n*–1 jobs, sequence them in the 2nd to *n*th positions in nondecreasing order of their growth rates. Stop.

Obviously, the complexity of Algorithm AGL1 is $O(n\log n)$. For Algorithm AGL1, utilizing the properties of Lemmas 1 and 2, we have the following conclusion.

Theorem 2. Algorithm AGL1 produces an optimal solution for the problem $1 \mid p_{i,r} = (p_0 + \alpha_i t) r^a, GL1 \mid C_{\max}$.

4.2 Problem $1 \mid p_{i,r} = (p_0 + \alpha_i t) r^a, GL2 \mid C_{\max}$

When the operator's learning ability follows a learning curve whose learning rate is between 50% and 100%, i.e., $-1 \le a < 0$, the jobs in positions 2 and 3 in an optimal schedule satisfy the following lemma.

Lemma 3. For the problem $1 \mid p_{i,r} = (p_0 + \alpha_i t) r^a, GL \mid C_{\max}$, if the learning index satisfies $-1 \le a < 0$, the growth rate of the job in position 2 should be no less than that of the job in position 3 in an optimal schedule.

Proof. For a schedule $\pi = [1, 2, \cdots, n]$, swap only the jobs in positions 2 and 3 to generated a new schedule π'. From (2), we have

$$C_{\max}(\pi') - C_{\max}(\pi) = p_0(\alpha_2 - \alpha_3)[(3^a - 2^a) + 2^a 3^a](1 + \alpha_4 4^a) \cdots (1 + \alpha_n n^a)$$

$$= p_0(\alpha_2 - \alpha_3) 2^a 3^a [1 - (3^{-a} - 2^{-a})](1 + \alpha_4 4^a) \cdots (1 + \alpha_n n^a).$$

From (6), we have

$$3^{-a} - 2^{-a} = (-a)(2+\theta)^{-a-1},$$

where $0 < \theta < 1$. If $-1 \le a < 0$, then $-1 < -a - 1 \le 0$. So, we have

$$3^{-a} - 2^{-a} = (-a)(2+\theta)^{-a-1} \le 1,$$

Therefore, if $-1 \le a < 0$ and $\alpha_2 \ge \alpha_3$, we have

$$C_{\max}(\pi') - C_{\max}(\pi) \ge 0. \qquad \square$$

Theorem 3. For the problem $1 \mid p_{i,r} = (p_0 + \alpha_i t) r^a, GL2 \mid C_{\max}$, the jobs should be sequenced in nonincreasing order of their growth rates to minimize the makespan.

Proof. According to Lemmas 1 and 3, the theorem holds for the first three positions in an optimal schedule.

In the following we show that the theorem also holds for the 3rd to nth positions in an optimal schedule. Assume that we are given a schedule $\pi = [1, 2, \cdots, n]$ and $\alpha_1 \ge \alpha_2 \ge \cdots \ge \alpha_n$. Swapping two jobs i ($i > 2$) and $i + 1$ to produce a new schedule π'. For the parameter distribution in $GL2$, making use of (2), we can derive the following inequality:

$$C_{\max}(\pi') - C_{\max}(\pi) \ge p_0(\alpha_i - \alpha_{i+1}) i^a (i+1)^a \left\{ 1 - [(i+1)^{-a} - i^{-a}] \frac{(1+1/n)^{i-1} - 1}{1/n} \right\}$$

$$(1 + \alpha_{i+2}(i+2)^a) \cdots (1 + \alpha_n n^a). \tag{8}$$

For the function $f(x) = (1+x)^{i-1}$, its derivative function $\frac{df(x)}{dx} = (i-1)(1+x)^{i-2}$. According to the Lagrange finite increment theorem, there exists a constant θ_1, $0 < \theta_1 < 1$, such that

$$\frac{(1+1/n)^{i-1} - 1}{1/n} = (i-1)\left(1 + \frac{\theta_1}{n}\right)^{i-2}.$$

From (6), there also exists a constant θ_2, $0 < \theta_2 < 1$, such that

$$(i+1)^{-a} - i^{-a} = (-a)(i+\theta_2)^{-a-1}.$$

Therefore, from (8), we have

$$C_{\max}(\pi') - C_{\max}(\pi) \geq p_0(\alpha_i - \alpha_{i+1}) i^a (i+1)^a \left\{ 1 - (-a)(i+\theta_2)^{-a-1}(i-1)\left(1+\frac{\theta_1}{n}\right)^{i-2} \right\}$$

$$(1+\alpha_{i+2}(i+2)^a)\cdots(1+\alpha_n n^a)$$

$$\geq p_0(\alpha_i - \alpha_{i+1}) i^a (i+1)^a \left\{ 1 - (-a)\frac{(i-1)(1+1/n)^{i-2}}{i^{1+a}} \right\}$$

$$(1+\alpha_{i+2}(i+2)^a)\cdots(1+\alpha_n n^a)$$

$$\geq p_0(\alpha_i - \alpha_{i+1}) i^a (i+1)^a \left\{ 1 - (-a)\frac{(1+1/n)^{i-2}}{i^{a}} \right\}$$

$$(1+\alpha_{i+2}(i+2)^a)\cdots(1+\alpha_n n^a)\,. \tag{9}$$

In $GL2$, $-\frac{1}{en} \leq a < 0$, then $0 < -a \leq \frac{1}{en}$. Consider $\left(1+\frac{1}{n}\right)^{i-2} < e$ for $3 \leq i \leq n$, therefore

$$(-a)\frac{(1+1/n)^{i-2}}{i^a} < \frac{i^{-a}}{n}\frac{(1+1/n)^{i-2}}{e} < 1\,.$$

Then, from (9), we have

$$C_{\max}(\pi') - C_{\max}(\pi) \geq 0. \qquad \square$$

According to Theorem 3, an algorithm that sequences the jobs in nonincreasing order of their growth rates produces an optimal solution for the problem $1 \mid p_{i,r} = (p_0 + \alpha_i t) r^a, GL2 \mid C_{\max}$. Obviously, the time complexity of the algorithm is $O(n\log n)$.

4.3 Problem $1 \mid p_{i,r} = (p_0 + \alpha_i t) r^a, GL3 \mid C_{\max}$

In $GL3$, the coefficient β indicates the constraints imposed on the distance of the growth rates between the jobs and the scope of the learning index. When β becomes larger, the distribution of the growth rates becomes more sparse and the learning index has a wider scope.

In this case, $GL3$ gives the boundaries on the parameter distribution of the growth rates involving n–2 jobs that are numbered as job $3, \cdots,$ job n. We derive an optimal property for these n–2 jobs in the following.

Lemma 4. For the problem $1 \mid p_{i,r} = (p_0 + \alpha_i t) r^a$, $GL3 \mid C_{\max}$, suppose the n–2 jobs, indexed as job $3, \cdots,$ job n, are determined in the last n–2 positions in a schedule, then these jobs should be sequenced in nondecreasing order of their growth rates to minimize the makespan.

Proof. For a schedule $\pi = [1, 2, \cdots, n]$, we only swap the jobs in positions i ($i > 2$) and i +1 to yield a new schedule π'. In GL3, making use of (2), we have

$$C_{\max}(\pi') - C_{\max}(\pi) \geq p_0(\alpha_{i+1} - \alpha_i) i^a (i+1)^a \left\{ [(i+1)^{-a} - i^{-a}](i-1)^a \frac{(1+\beta)^{i-1} - 1}{\beta} - 1 \right\}$$

$$(1 + \alpha_{i+2}(i+2)^a) \cdots (1 + \alpha_n n^a) . \qquad (10)$$

For $i \geq 3$, we will prove that the following inequality holds:

$$[(i+1)^{-a} - i^{-a}](i-1)^a \frac{(1+\beta)^{i-1} - 1}{\beta} - 1 > 0 . \qquad (11)$$

Obviously, we have

$$\frac{(1+\beta)^{i-1} - 1}{\beta} > \frac{(1 + (i-1)\beta + \beta^2 (i-1)(i-2)/2) - 1}{\beta} = (i-1) + \frac{(i-1)(i-2)}{2}\beta. \qquad (12)$$

From (6), we have

$$(i+1)^{-a} - i^{-a} = \frac{(-a)}{(i+\theta)^{1+a}} > \frac{(-a)}{(i+1)^{1+a}}. \qquad (13)$$

where $0 < \theta < 1$.

Then, from (12) and (13), we have

$$[(i+1)^{-a} - i^{-a}](i-1)^a \frac{(1+\beta)^{i-1} - 1}{\beta} > (-a)\left(\frac{i-1}{i+1}\right)^{1+a} \left(1 + \frac{i-2}{2}\beta\right). \qquad (14)$$

In GL3, $a < -\dfrac{4}{4+\beta}$, then

$$a < -\frac{4}{4+\beta} \le -\frac{1}{1+\frac{(i-1)(i-2)}{2(i+1)}\beta}, \text{ for } i = 3, 4, \cdots, n-1,$$

i.e.,

$$(-a)\left(1+\frac{(i-1)(i-2)}{2(i+1)}\beta\right) > 1,$$

then,

$$(-a)\left(\frac{i+1}{i-1}\right)\left(1+\frac{(i-1)(i-2)}{2(i+1)}\beta\right) > \frac{i+1}{i-1}.$$

That is,

$$(-a)\left(\frac{i+1}{i-1}+\frac{i-2}{2}\beta\right) > 1+\frac{2}{i-1},$$

i.e.,

$$(-a)\left(1+\frac{i-2}{2}\beta\right) > 1+\frac{2}{i-1}(1+a). \tag{15}$$

Notice the Bernoulli inequality: if $0 < 1+a < 1$, then

$$\left(\frac{i+1}{i-1}\right)^{1+a} = \left(1+\frac{2}{i-1}\right)^{1+a} \le 1+\frac{2}{i-1}(1+a). \tag{16}$$

From (15) and (16), we have

$$(-a)\left(1+\frac{i-2}{2}\beta\right) > \left(\frac{i+1}{i-1}\right)^{1+a},$$

i.e.,

$$(-a)\left(\frac{i-1}{i+1}\right)^{1+a}\left(1+\frac{i-2}{2}\beta\right) > 1. \tag{17}$$

From (14) and (17), we have shown that (11) holds. Thus, from (10), we have

$$C_{\max}(\pi') - C_{\max}(\pi) \ge 0. \tag{18}$$

Since (18) holds for $i = 3, 4, \cdots, n-1$, we have reached the conclusion. □

Suppose that the schedule $\pi = [1, 2, \cdots, n]$ is an optimal solution for the problem $1 \mid p_{i,r} = (p_0 + \alpha_i t)r^a, GL3 \mid C_{\max}$. According to Lemmas 3 and 4, both

$\alpha_2 \ge \alpha_3$ and $\alpha_3 \le \alpha_4 \le \cdots \le \alpha_n$ hold. In the following we derive an optimal property relating to the relations between α_2 and $\alpha_4, \alpha_5, \cdots, \alpha_n$.

Lemma 5. For the problem $1 \mid p_{i,r} = (p_0 + \alpha_i t) r^a, GL3 \mid C_{\max}$, if the schedule $\pi = [1, 2, \cdots, n]$ is an optimal solution, then $\alpha_2 \le \alpha_i$ if $i^{-a} \ge 2^{-a} + 1 + \frac{1}{\beta} - \frac{1}{\beta(1+\beta)^{i-3}}$.

Proof. We assume that schedule π' is obtained by swapping the 2nd and *i*th jobs ($i > 3$) in schedule π. Since $\alpha_3 \ge 3^{-a}\beta$ and $\alpha_{i+1} - \alpha_i \ge [(i+1)^{-a} - i^{-a}]\beta$, for $i = 3, 4, \cdots, n-1$, in *GL*3, making use of (2), we have

$$
\begin{aligned}
C_{\max}(\pi') - C_{\max}(\pi) \ge{} & p_0(\alpha_i - \alpha_2)\{(2^a - i^a)(1+\beta)^{i-3} - i^a[2^a(1+\beta)^{i-3} + \cdots + \\
& (i-2)^a(1+\beta) + (i-1)^a]\}(1+\alpha_{i+1}(i+1)^a)\cdots(1+\alpha_n n^a) \\
\ge{} & p_0(\alpha_i - \alpha_2)\{(2^a - i^a)(1+\beta)^{i-3} - i^a 2^a[(1+\beta)^{i-3} + \cdots + \\
& (1+\beta) + 1]\}(1+\alpha_{i+1}(i+1)^a)\cdots(1+\alpha_n n^a) \\
={} & p_0(\alpha_i - \alpha_2)2^a i^a\left\{(1+\beta)^{i-3}(i^{-a} - 2^{-a}) - \frac{(1+\beta)^{i-2} - 1}{\beta}\right\} \\
& (1+\alpha_{i+1}(i+1)^a)\cdots(1+\alpha_n n^a) \\
={} & (\alpha_i - \alpha_2)2^a i^a(1+\beta)^{i-3}\left\{\left(i^{-a} - 2^{-a}\right) - \left(1 + \frac{1}{\beta}\right) + \frac{1}{\beta(1+\beta)^{i-3}}\right\} \\
& (1+\alpha_{i+1}(i+1)^a)\cdots(1+\alpha_n n^a).
\end{aligned}
$$

Therefore, if

$$\alpha_i \ge \alpha_2 \text{ and } i^{-a} \ge 2^{-a} + \left(1 + \frac{1}{\beta}\right) - \frac{1}{\beta(1+\beta)^{i-3}},$$

then $C_{\max}(\pi') - C_{\max}(\pi) \ge 0$. □

Making use of Lemmas 1, 3, 4 and 5, we can develop an algorithm for the problem $1|p_{i,r} = (p_0+\alpha_i t)r^a, GL3|C_{\max}$. The algorithm is described in detail in the following.

Algorithm AGL3:

Step 1. Select a job with the largest growth rate, and sequence it in the first position.

Step 2. For the remaining $n-1$ jobs, number them such that $\alpha_2 \le \alpha_3 \le \cdots \le \alpha_n$, where α_i is the growth rate of job i, $i = 2, 3, \cdots, n$.

Determine an integer i_1 such that $i_1 = \min\{i|i^{-a} \ge 2^{-a}+1+1/\beta\}$.

For $k = 3$ to i_1-1:

Sequence job k in the 2nd position, and the remaining $n-2$ jobs in nondecreasing order of their growth rates, i.e., $\alpha_2 \le \cdots \le \alpha_{k-1} \le \alpha_{k+1} \le \cdots \le \alpha_n$. Compute the objective function as $C_{\max}(k)$.

Step 3. Select j such that $C_{\max}(j) = \min\{C_{\max}(3), C_{\max}(4), \cdots, C_{\max}(i_1-1)\}$. Then, determine the schedule with the makespan $C_{\max}(j)$. Stop.

In Algorithm AGL3, the dominant computational step is to sequence the jobs in nondecreasing order of their growth rates, which requires $O(n\log n)$ times. So the time complexity of Algorithm AGL3 is $O(n\log n)$.

According to Lemma 1, in Step 1, we determine a job that should be sequenced in the first position in an optimal solution. For the remaining $n-1$ positions in an optimal solution, Steps 2 and 3 ensure that the conditions of Lemmas 3, 4 and 5 are satisfied. So Algorithm AGL3 generates an optimal solution. Thus, we have established the following theorem.

Theorem 4. For the problem $1|p_{i,r} = (p_0+\alpha_i t)r^a, GL3|C_{\max}$, Algorithm AGL3 yields an optimal schedule.

5. A special model with a bounded heuristic

In this section we introduce a special model with a parameter subset $GL4$. $GL4$ is defined as

$$GL4 = \Big\{\{\alpha_1, \cdots, \alpha_n\}, \{a\} \Big| \alpha_1 \geq \max\{\alpha_2, \cdots, \alpha_n\}, \left|\alpha_i - \alpha_j\right| \leq \frac{2^{-a}}{n}$$

$$\text{for } \; i \neq j \text{ and } \; i, j \geq 2, \;\; -1 < a < 0\Big\}.$$

We denote the general model with the parameter subset $GL4$ as $1 | p_{i,r} = (p_0 + \alpha_i t)r^a, GL4 | C_{\max}$.

The parameter distribution of the growth rates and learning index in $GL4$ reflects the actual scheduling situation in which the deterioration effects on all the jobs have little differences and the learning process to increase operating efficiency is not very fast.

We notice that $GL2 \subset GL4$, which indicates that this case models more general practical scheduling situations than $1 | p_{i,r} = (p_0 + \alpha_i t)r^a, GL2 | C_{\max}$. Consequently, the LGR rule schedule, which is optimal for the problem $1 | p_{i,r} = (p_0 + \alpha_i t)r^a, GL2 | C_{\max}$, is no longer optimal for this case. However, we show in the following that the LGR rule schedule is bounded for the problem $1 | p_{i,r} = (p_0 + \alpha_i t)r^a, GL4 | C_{\max}$.

Lemma 6. Suppose the n–m+1 numbers $\alpha_m, \alpha_{m+1}, \cdots, \alpha_n$ satisfy $0 < \alpha_m \leq \alpha_{m+1} \leq \cdots \leq \alpha_n$, and $\alpha_{i_m}, \alpha_{i_{m+1}}, \cdots, \alpha_{i_n}$ are obtained by re-indexing $\alpha_m, \alpha_{m+1}, \cdots, \alpha_n$, and a < 0, then

$$\prod_{k=m}^{n}(1 + k^a \alpha_{i_k}) \geq \prod_{k=m}^{n}(1 + k^a \alpha_k). \tag{19}$$

Proof. We use induction to prove the result.

The case $n = m + 1$:

If $\alpha_{i_m} = \alpha_m$, $\alpha_{i_{m+1}} = \alpha_{m+1}$, then (19) holds. Otherwise, we have

$$(1+m^a\alpha_{i_m})(1+(m+1)^a\alpha_{i_{m+1}}) = (1+m^a\alpha_m)(1+(m+1)^a\alpha_{m+1})$$
$$+(m^a-(m+1)^a)(\alpha_{m+1}-\alpha_m)\,.$$

Therefore,

$$(1+m^a\alpha_{i_m})(1+(m+1)^a\alpha_{i_{m+1}}) \geq (1+m^a\alpha_m)(1+(m+1)^a\alpha_{m+1}).$$

Suppose (19) holds for the case *n*, and $[\alpha_{i_m}, \alpha_{i_{m+1}}, \cdots, \alpha_{i_n}, \alpha_{i_{n+1}}]$ is a permutation of $\alpha_m, \alpha_{m+1}, \cdots, \alpha_n, \alpha_{n+1}$.

If $\alpha_{i_{n+1}} = \alpha_{n+1}$, then by the induction principle, we have

$$\prod_{k=m}^{n+1}(1+k^a\alpha_{i_k}) = (1+(n+1)^a\alpha_{n+1})\prod_{k=m}^{n}(1+k^a\alpha_{i_k}) \geq \prod_{k=m}^{n+1}(1+k^a\alpha_k)\,.$$

Otherwise, there exists a number α_{i_j} such that $\alpha_{i_j} = \alpha_{n+1}$. Then, we have

$$\prod_{k=m}^{n+1}(1+k^a\alpha_{i_k}) = (\prod_{k=m}^{j-1}(1+k^a\alpha_{i_k}))(1+j^a\alpha_{i_{n+1}})(\prod_{k=j+1}^{n}(1+k^a\alpha_{i_k}))(1+(n+1)^a\alpha_{n+1})$$
$$+(j^a-(n+1)^a)(\alpha_{n+1}-\alpha_{i_{n+1}})(\prod_{k=m}^{j-1}(1+k^a\alpha_{i_k}))(\prod_{k=j+1}^{n}(1+k^a\alpha_{i_k}))\,.$$

So, we have

$$\prod_{k=m}^{n+1}(1+k^a\alpha_{i_k}) \geq \prod_{k=m}^{n+1}(1+k^a\alpha_k)\,.$$

Hence, we have proved that (19) holds for the case $n+1$.

Thus, according to the induction principle, we have established the result. □

Making use of Lemma 6, we can give a lower bound for $1 \mid p_{i,r} = (p_0+\alpha_i t)r^a,\ GL \mid C_{\max}$ as follows.

Lemma 7. For the problem $1 \mid p_{i,r} = (p_0+\alpha_i t)r^a,\ GL \mid C_{\max}$, if the *n* jobs are numbered such that $\alpha_1 \ge \alpha_2 \ge \cdots \ge \alpha_n$, then the makespan $C_{\max}(\pi)$ of any schedule π satisfies

$$C_{\max}(\pi) \ge p_0 \sum_{k=1}^{n} [k^a \prod_{i=k+1}^{n} (1 + \alpha_i (n - i + k + 1)^a)], \tag{20}$$

where $\prod_{i=n+1}^{n} (1 + \alpha_i (n - i + k + 1)^a) := 1$.

Proof. For any schedule $\pi = [i_1, i_2, \cdots, i_n]$, from (2), its makespan is

$$C_{\max}(\pi) = p_0 \sum_{k=1}^{n} [k^a \prod_{j=k+1}^{n} (1 + \alpha_{i_j} j^a)].$$

For $k = 1, 2, \cdots, n-1$, re-number $\alpha_{i_{k+1}}, \alpha_{i_{k+2}}, \cdots, \alpha_{i_n}$ as $\alpha'_{k+1}, \cdots, \alpha'_n$ such that $\alpha'_{k+1} \ge \cdots \ge \alpha'_n$. From Lemma 6, we have

$$\prod_{j=k+1}^{n} (1 + \alpha_{i_j} j^a) \ge \prod_{j=k+1}^{n} (1 + \alpha'_j (n - j + k + 1)^a).$$

Since $\alpha_1 \ge \alpha_2 \ge \cdots \ge \alpha_n$, obviously

$$\prod_{j=k+1}^{n} (1 + \alpha'_j (n - j + k + 1)^a) \ge \prod_{j=k+1}^{n} (1 + \alpha_j (n - j + k + 1)^a).$$

Therefore,

$$\prod_{j=k+1}^{n} (1 + \alpha_{i_j} j^a) \ge \prod_{j=k+1}^{n} (1 + \alpha_j (n - j + k + 1)^a).$$

So, we have

$$C_{\max}(\pi) \ge p_0 \sum_{k=1}^{n} [k^a \prod_{j=k+1}^{n} (1 + \alpha_j (n - j + k + 1)^a)]. \qquad \square$$

For the parameter distribution of growth rates and learning index in *GL*4, we denote the LGR rule schedule and the optimal schedule as π and π^*, respectively. The LGR rule schedule has the following performance bound.

Theorem 5. For the problem $1 | p_{i,r} = (p_0 + \alpha_i t) r^a,\ GL4 | C_{\max}$, the LGR rule schedule has a performance bound

$$C_{\max}(\pi) / C^*_{\max} < e^{2^{-a-1}},$$

where e is the base of the natural logarithm.

Proof. For schedule $\pi = [1, 2, \cdots, n]$ that follows the LGR rule, its makespan is

$$C_{\max}(\pi) = p_0 \sum_{k=1}^{n} [k^a \prod_{i=k+1}^{n} (1 + \alpha_i i^a)]. \tag{21}$$

Comparing the objective function (21) of the schedule π and its lower bound (20) in $GL4$, we have

$$\frac{(1+\alpha_n n^a)}{(1+\alpha_n n^a)} = 1,$$

$$\frac{(1+\alpha_{n-1}(n-1)^a)(1+\alpha_n n^a)}{(1+\alpha_n (n-1)^a)(1+\alpha_{n-1} n^a)} \le \frac{1+\alpha_{n-1}(n-1)^a}{1+\alpha_n (n-1)^a} = 1 + \frac{(\alpha_{n-1} - \alpha_n)(n-1)^a}{1+\alpha_n (n-1)^a}$$

$$\le 1 + \frac{2^{-a}}{n},$$

$$\frac{(1+\alpha_{n-2}(n-2)^a)(1+\alpha_{n-1}(n-1)^a)(1+\alpha_n n^a)}{(1+\alpha_n (n-2)^a)(1+\alpha_{n-1}(n-1)^a)(1+\alpha_{n-2} n^a)} \le \frac{1+\alpha_{n-2}(n-2)^a}{1+\alpha_n (n-2)^a}$$

$$= 1 + \frac{(\alpha_{n-2} - \alpha_n)(n-2)^a}{1+\alpha_n (n-2)^a} < 1 + \frac{2^{-a}}{n}.$$

Generally, for $k = n–4, n–5, \cdots, 1$, we have

$$\prod_{i=k+1}^{n} (1+\alpha_i i^a) / \prod_{i=k+1}^{n} (1+\alpha_i (n-i+k+1)^a) < \left(1 + \frac{2^{-a}}{n}\right)^{\left\lfloor \frac{n-k+1}{2} \right\rfloor},$$

where $\lfloor x \rfloor$ denotes the largest integer that is less than x. Therefore,

$$C_{\max}(\pi) / \left\{ p_0 \sum_{k=1}^{n} [k^a \prod_{i=k+1}^{n} (1+\alpha_i (n-i+k+1)^a)] \right\}$$

$$\le \max\left\{ \left(1+\frac{2^{-a}}{n}\right)^{\left\lfloor \frac{n-k+1}{2} \right\rfloor} \middle| k = n-1, n-2, \cdots, 1 \right\}. \qquad (22)$$

From (22), we have

$$C_{\max}(\pi)/C^*_{\max} < \left(1+\frac{2^{-a}}{n}\right)^{\frac{n}{2}} < e^{2^{-a-1}}.$$

That is

$$C_{\max}(\pi)/C^*_{\max} < e^{2^{-a-1}}. \qquad \square$$

In Theorem 5 the performance bound is related to the learning index a. In $GL4$, $a \in (-1, 0)$. So, the performance bounds are between $\sqrt{e}$ and e, where e is the base of the natural logarithm. However, if we make the distances of the growth rates of the jobs shorter, then the performance bound will become tighter.

6. Conclusion

In this paper we studied the simultaneous effects of deterioration and learning on single-machine scheduling to minimize the makespan. For the general model, its inherent complexities make the LGR rule to be unbounded, although the LGR rule yields an optimal solution for the scheduling problem considering deteriorating jobs only. We modeled some practical scheduling scenarios, and developed optimal algorithms for them based on the derived optimal properties. Finally, we focused on a special model and showed that there exists a bounded heuristic for it.

Acknowledgements

We are grateful for two anonymous referees for their helpful comments on an earlier version of this paper. This research was supported in part by The Hong Kong Polytechnic University under a grant from the *Area of Strategic Development in Chain Business Services*.

References

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

Bachman, A. and Janiak, A. (2000) “Minimizing maximum lateness under linear deterioration,” *European Journal of Operational Research*, 126, 557-566.

Bachman, A., Janiak, A. and Kovalyov, M. Y. (2002) “Minimizing the total weighted completion time of deteriorating jobs,” *Information Processing Letters*, 81, 81-84.

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

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

Cheng, T. C. E., Ding, Q. and 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. and Kovalyov, M. Y. (1994) “Scheduling with learning effects on job processing times,” *Working Paper, No. 06/94*, Faculty of Business and Information Systems, The Hong Kong Polytechnic University.

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

Graham, R. L., Lawler, E. L., Lenstra, J. K. and Rinnooy Kan, A. H. G.. (1979) “Optimization and approximation in deterministic sequencing and scheduling: a survey,” *Annals of Discrete Mathematics*, 5, 287-326.

Kunnathur, A. S. and 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.

Mosheiov, G.. (1991) “V-shaped policies for scheduling deteriorating jobs,” *Operations Research*, 39, 979-991.

Mosheiov, G.. (1994) “Scheduling deteriorating jobs under simple linear deterioration,” *Computers and Operations Research*, 21, 653-659.

Mosheiov, G.. (1995) “Scheduling jobs with step-deterioration: minimizing makespan on a single- and multi-machine,” *Computers and Industrial Engineering*, 28, 869-879.

Mosheiov, G.. (2001) “Scheduling problems with a learning effect,” *European Journal of Operational Research*, 132, 687-693.

Wang, J-B. and Cheng, T. C. E. (2005) “Scheduling problems with the effects of deterioration and learning,” *Asia-Pacific Journal of Operational Research*, accepted.