Scheduling problems with the effects of deterioration and learning
(Revised version of APJOR ♯ 041101)

Ji-Bo Wang[a,b,1] T.C.E. Cheng[b]
[a]Department of Science, Shenyang Institute of Aeronautical Engineering,
Shenyang 110034, People's Republic of China
[b]Department of Logistics, The Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong SAR, People's Republic of China

Abstract

This paper deals with the machine scheduling problems with the effects of deterioration and learning. In this model, the processing times of the jobs are defined as functions of their starting times and positions in a sequence. We introduce polynomial solutions for some single machine problems and flow shop problems. The performance measures include makespan, total completion time, total weighted completion time, and maximum lateness.

Keywords: Scheduling, Single machine, Flow shop, Learning effect, Deteriorating jobs

1 Introduction

In the classical scheduling theory, the job processing times are considered to be constant. In practice, however, we often encounter settings in which the job processing times may be subject to change due to the phenomenon of learning or deterioration (Pinedo [28]). For example, when the processing times arise from manual operations, the possibility of learning exists. Biskup [7] indicated that the learning effect has been observed in numerous practical situations in different sectors of industry and for a variety of corporate activities. On the other hand, it has been noticed that jobs may deteriorate as they wait to be processed. Kunnathur and Gupta [16] and Mosheiov [20] presented several real-life situations where deteriorating jobs might occur. Lee [17] first considered the effects of deterioration and learning simultaneous. The phenomena of learning effect and deteriorating jobs occurring simultaneously can be found in many real-life situations. For example, as the manufacturing environment becomes increasingly competitive, in order to provide customers with greater product variety, organizations are moving towards shorter production runs and frequent product changes. 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.

[1] Corresponding author. E-mail addresses: wangjibo75@yahoo.com.cn; lgtcheng@polyu.edu.hk

These workers are often interrupted by product and process changes causing deterioration in performance, which we will refer to, for simplicity, as forgetting. Considering learning and forgetting effects in measuring productivity should be helpful in improving the accuracy of production planning and productivity estimation (Nembhard and Osothsilp [26]). In this paper we investigate the implications of these phenomena occurring simultaneously for single machine and flow shop scheduling problems.

Analysis of scheduling problems in which the processing time of a job is an increasing function of its starting time was introduced by Browne and Yechiali [8]. Mosheiov [18] considered the flow time minimization problem under the assumption that basic processing times remain the same in the linear deterioration model. The motivation for analyzing identical basic processing times arises not only from its intrinsic interest, but it also serves as a good approximation to the general case. Later, Mosheiov [19] further simplified the model to a simple linear deterioration model in which the jobs have a fixed job-dependent growth rate but no basic processing times. This follows from the fact that as the number of jobs increases, the starting times of many jobs are postponed and their basic processing times become irrelevant. Sundararaghavan and Kunnathur [29] considered the single machine scheduling problem in which the processing time is a binary function of a common start time due date. The jobs have processing time penalties for starting after the due date, and the objective was to minimize the sum of the weighted completion times. Three special cases of this problem can be solved optimally. Bachman and Janiak [3] showed that the maximum lateness minimization problem under the linear deterioration assumption is NP-hard, and two heuristic algorithms are presented as a consequence. Bachman et al. [5] considered the problem of minimizing the total weighted completion time introduced by Browne and Yechiali [8]. They proved that the problem is NP-hard.

Chen [9] and Mosheiov [21] considered scheduling deteriorating jobs in a *multi-machine* setting. They assumed linear deterioration and parallel identical machines. Chen considered the minimum flow time and Mosheiov studied makespan minimization. Mosheiov [22] considered the complexity of flow shop, open shop and job shop makespan minimization problems. Mosheiov introduced a polynomial-time algorithm for the two-machine flow shop and proved its NP-hardness when an arbitrary number of machines (three or more) are assumed. Wang and Xia [34] considered no-wait or no-idle flow shop scheduling problems with job processing times dependent on their starting times. In these problems the job processing time is a simple linear function of a job's starting time and some dominating relationships between machines can be satisfied. They showed that for the problems to minimize makespan or weighted sum of completion times, polynomial algorithms still exist. When the objective is to minimize maximum lateness, the solutions of a classical version may not hold. Other types of deterioration have also been discussed. For instance, Kunnathur and Gupta [16], and Mosheiov [20] considered piecewise linear deteriorating functions.

Apart from the increasing linear model for the job processing times, there is also a decreasing linear model, which essentially presents the learning effect from modelling aspect. This model was introduced by Ho et al. [15]. Ho et al. [15] considered the problem of solution feasibility with deadline restrictions. Cheng and Ding [10] considered some problems with an increasing/decreasing linear model of the job processing times, but with ready time and deadline restrictions. They identified some interesting relationships between the linear models with decreasing and increasing start time dependent parts. Ng et al. [27] considered three scheduling problems with a decreasing linear model of the job processing times, where the objective function was to minimize the total completion time, and two of the problems are solved optimally. A pseudopolynomial time algorithm was constructed to solve the third problem using dynamic programming. Some interesting relationships between the linear model with decreasing and increasing start time dependent parts have also been presented by Ng et al. [27]. Bachman et al. [2] considered the single machine scheduling problem with start time dependent job processing times. They proved that the problem of minimizing the total weighted completion time is NP-hard. They also considered some special cases. Wang and Xia [32] considered the scheduling problems under a special type of linear decreasing deterioration. They presented optimal algorithms for single machine scheduling of minimizing the makespan, maximum lateness, maximum cost and number of late jobs, respectively. For the two-machine flow shop scheduling problem to minimize the makespan, they proved that the optimal schedule can be obtained by Johnson's rule. If the processing times of the operations are equal for each job, they proved that the flow shop scheduling problems can be transformed into single machine scheduling problems. Extensive reviews of research on scheduling deteriorating jobs have been provided by Alidaee and Womer [1] and Cheng et al. [11].

Biskup [7] and Cheng and Wang [12] were among the pioneers that brought the concept of learning into the field of scheduling, although it has been widely employed in management science since its discovery by Wright [35]. Biskup [7] proved that single-machine scheduling with a learning effect remains polynomial solvable if the objective is to minimize the deviation from a common due date or to minimize the sum of flow time. Cheng and Wang [12] considered a single machine scheduling problem in which the job processing times will decrease as a result of learning. A volume-dependent piecewise linear processing time function was used to model the learning effects. The objective was to minimize the maximum lateness. They showed that the problem is NP-hard in the strong sense and then identified two special cases which are polynomially solvable. They also proposed two heuristics and analysed their worst-case performance. Mosheiov [23, 24] investigated several other single-machine problems, and the problem of minimizing the total flow time on identical parallel machines. Mosheiov and Sidney

[25] considered the case of a job-dependent learning curve, where the learning in the production process of some jobs is faster than that of others. Wang and Xia [33] considered flow shop scheduling problems with a learning effect. The objective was to minimize one of two regular performance criteria, namely makespan and total flow time. They gave a heuristic algorithm with a worst-case error bound of m for each criterion, where m is the number of machines. They also found polynomial time solutions to two special cases of the problems, i.e., identical processing times on each machine and an increasing series of dominating machines. A survey on this line of the scheduling research could be found in Bachman and Janiak [4].

In this paper we study scheduling problems with the effects of deterioration and learning. The remaining part of this paper is organized as follows. In section 2 we consider single machine scheduling problems. We show that for some special cases, the solutions of the classical versions also hold for the versions with the effects of deterioration and learning. In section 3 we consider flow shop scheduling problems and show that for some special cases the problems can be solved. The last section is the conclusion.

2 Single machine problems

The focus of this paper is to study the effects of deterioration and learning simultaneously. The learning effect model provided by Biskup [7] is combined with the linear deterioration model in which the basic job processing time is proportional to the deteriorating rate to yield our model. The model is described as follows.

The single machine problem is to schedule n jobs $J_1, J_2, ..., J_n$ on one machine. All the jobs are available for processing at some time $t_0 \geq 0$. The machine can handle one job at a time and preemption is not allowed. Associated with each job $J_j (j = 1, 2, ..., n)$ is a weight w_j and a due date d_j. Let $p_{j,r}(t)$ be the processing time of job J_j if it is started at time t and scheduled in position r in a sequence. The general model is

$$p_{j,r}(t) = (p_j + \alpha_j t) r^a,$$

where p_j is the basic processing time of the job J_j, i.e., the processing time of a job if it is scheduled first in a sequence and its starting time is 0, i.e., t=0 and r=1, α_j is its deterioration rate and $a \leq 0$ is its learning index. Lee [17] considered the models where the processing times are $p_{j,r}(t) = \alpha_j t r^a$ and $p_{j,r}(t) = (p_0 + \alpha_j t) r^a$, where α_j is the deterioration rate of job J_j, p_0 is the common basic processing time and $a \leq 0$ is the learning index, given as the (base 2) logarithm of the learning rate [6]. In this paper we consider a new model where $p_j = b\alpha_j$ and $p_{j,r} = \alpha_j (b + t) r^a$. In fact, we consider the following general model

$$p_{j,r}(t) = \alpha_j (b + ct) r^a. \tag{1}$$

Obviously, when $b = 0, c = 1$, model (1) is the model Lee [17], when $b = 1, c = 0$, model (1) is the model of Biskup [7] and Mosheiov [23].

For a given schedule $\pi = [J_1, J_2, \ldots, J_n]$, $C_j = C_j(\pi)$ represents the completion time of job J_j and $f(C) = f(C_1, C_2, \ldots, C_n)$ is a regular measure of performance. Let $C_{\max} = \max\{C_j | j = 1, 2, \ldots, n\}$, $\sum C_j$, $\sum w_j C_j$ and $L_{\max} = \max\{C_j - d_j | j = 1, 2, \ldots, n\}$ represents the makespan, sum of completion times, weighted sum of completion times and maximum lateness of a given permutation, respectively. In the remaining part of the paper, all the problems considered will be denoted using the three-field notation schema $\alpha|\beta|\gamma$ introduced by Graham et al. [13].

In this section we examine several well-known classical single-machine scheduling problems under the assumption that the actual processing time has the form of the model (1).

Lemma 1 *For a given scheduling $\pi = [J_1, J_2, \ldots, J_n]$ of $1|\alpha_j(b + ct)r^a|f(C)$, if job J_j starts at time $t_0 \geq 0$, then its completion time C_j is equal to*

$$C_j = (t_0 + \frac{b}{c})\prod_{i=1}^{j}(1 + c\alpha_i i^a) - \frac{b}{c}. \tag{2}$$

Proof: (by induction).

$$C_1 = t_0 + \alpha_1(b + ct_0)1^a = (t_0 + \frac{b}{c})(1 + c\alpha_1 1^a) - \frac{b}{c},$$

$$C_2 = C_1 + \alpha_2(b + cC_1)2^a = (t_0 + \frac{b}{c})(1 + c\alpha_1 1^a)(1 + c\alpha_2 2^a) - \frac{b}{c}.$$

Suppose Lemma 1 holds for job J_j, i.e.,

$$C_j = (t_0 + \frac{b}{c})\prod_{i=1}^{j}(1 + c\alpha_i i^a) - \frac{b}{c}.$$

Consider job J_{j+1}.

$$C_{j+1} = C_j + \alpha_{j+1}(b + cC_j)(j + 1)^a = (t_0 + \frac{b}{c})\prod_{i=1}^{j+1}(1 + c\alpha_i i^a) - \frac{b}{c}.$$

Hence, Lemma 1 holds for J_{j+1}. This completes the proof of Lemma 1. □

Theorem 1 *For the problem $1|\alpha_j(b + ct)r^a|C_{\max}$, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of α_j (i.e., the smallest deterioration rate (SDR) rule).*

Proof: Suppose $\alpha_i \leq \alpha_j$. Let π and π' be two job schedules where the difference between π and π' is the pairwise interchange of two adjacent jobs J_i and J_j, that is, $\pi = [S_1, J_i, J_j, S_2], \pi' =$

$[S_1, J_j, J_i, S_2]$, where S_1 and S_2 are partial sequences. Furthermore, we assume that there are $r-1$ jobs in S_1. Thus, J_i and J_j are the rth and the $(r+1)$th jobs, respectively, in π. Likewise, J_j and J_i are scheduled in the rth and the $(r+1)$th positions in π'. To further simplify the notation, let t denote the completion time of the last job in S_1. To show π dominates π', it suffices to show that the $(r+1)$th jobs in π and π' satisfy the condition that $C_j(\pi) \le C_i(\pi')$. The actual processing time of J_i in π is $p_{ir} = \alpha_i(b+ct)r^a$ and its completion time is

$$C_i(\pi) = (t + \frac{b}{c})(1 + c\alpha_i r^a) - \frac{b}{c}.$$

Thus, the actual processing time for J_j in π is $p_{j,r+1} = \alpha_j(b + cC_i(\pi))(r+1)^a$ and its completion time is

$$C_j(\pi) = C_i(\pi) + \alpha_j(b + cC_i(\pi))(r+1)^a = (t + \frac{b}{c})(1 + c\alpha_i r^a)(1 + c\alpha_j (r+1)^a) - \frac{b}{c}. \quad (3)$$

Similarly, it is easy to derive the completion times of J_j and J_i in π' as

$$C_j(\pi') = (t + \frac{b}{c})(1 + c\alpha_j r^a) - \frac{b}{c}$$

and

$$C_i(\pi') = (t + \frac{b}{c})(1 + c\alpha_j r^a)(1 + c\alpha_i (r+1)^a) - \frac{b}{c}. \quad (4)$$

Based on (3) and (4), we have

$$C_i(\pi') - C_j(\pi) = (ct + b)(\alpha_j - \alpha_i)(r^a - (r+1)^a) \ge 0.$$

Thus, π dominates π'. □

Theorem 2 *For the problem $1|\alpha_j(b+ct)r^a|\sum C_j$, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of α_j (the SDR rule).*

Proof: Similar to the proof of Theorem 1, except that

$$C_i(\pi') + C_j\pi') - C_i(\pi) - C_j(\pi) = (ct + b)(\alpha_j - \alpha_i)(2r^a - (r+1)^a) \ge 0.$$

□

For the other three objective functions, minimizing the total weighted completion time, minimizing maximum lateness, and minimizing the number of tardy jobs, Lee [17] showed that the $O(n \log n)$ solutions of these classical versions do not hold with the effects of deterioration and learning, hence for the model (1), the classical versions do not hold with the effects of deterioration and learning. But for some special cases, the problems can be solved in polynomial time.

Theorem 3 *For the problem* $1|\alpha_j(b+ct)r^a|\sum w_jC_j$*, if the jobs have agreeable weights, i.e.,* $\alpha_j \leq \alpha_k$ *implies* $w_j \geq w_k$ *for all the jobs* J_j *and* J_k*, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* α_j/w_j *(i.e., the weighted smallest deterioration rate (WSDR) rule).*

Proof: (by contradiction). Consider an optimal schedule π that does not follow the WSDR rule. In this schedule there must be at least two adjacent jobs, say job J_i followed by job J_j, such that $\alpha_i/w_i > \alpha_j/w_j$ (it implies $\alpha_i \geq \alpha_j$). Assume that job J_i is scheduled in position r and starts its processing at time t. Perform an adjacent pair-wise interchange of jobs J_i and J_j. Whereas under the original schedule π, job J_i is scheduled in position r and job J_j is scheduled in position $r+1$, under the new schedule job J_j is scheduled in position r and job J_i is scheduled in position $r+1$. All other jobs remain in their original positions. Call the new schedule π'. The completion times of the jobs processed before jobs J_i and J_j are not affected by the job interchange. Furthermore, the completion times of the jobs processed after jobs J_i and J_j will not increase by the interchange (since $\alpha_i \geq \alpha_j$). Under π,

$$C_i(\pi) = (t+\frac{b}{c})(1+c\alpha_i r^a) - \frac{b}{c}$$

$$C_j(\pi) = (t+\frac{b}{c})(1+c\alpha_i r^a)(1+c\alpha_j(r+1)^a) - \frac{b}{c}.$$

Whereas under π', they are

$$C_j(\pi') = (t+\frac{b}{c})(1+c\alpha_j r^a) - \frac{b}{c}$$

$$C_i(\pi') = (t+\frac{b}{c})(1+c\alpha_j r^a)(1+c\alpha_i(r+1)^a) - \frac{b}{c}.$$

So we have

$$\begin{aligned} & \sum w_jC_j(\pi') - \sum w_jC_j(\pi) \\ = \; & (ct+b)[(\alpha_j-\alpha_i)(r^a-(r+1)^a)(w_i+w_j) + (\alpha_j w_i - \alpha_i w_j)(r+1)^a \\ & + c(w_i-w_j)\alpha_i\alpha_j r^a(r+1)^a]. \end{aligned}$$

Since $\alpha_i \geq \alpha_j$, $r^a \geq (r+1)^a$, $w_j \geq w_i$ and $\alpha_i/w_i > \alpha_j/w_j$, then $\sum w_jC_j(\pi') - \sum w_jC_j(\pi) < 0$. It follows that the weighted sum of completion times under π' is strictly less than under π. This contradicts the optimality of π and proves the theorem. □

Theorem 4 *For the problem* $1|\alpha_j(b+ct)r^a|L_{\max}$*, if the jobs have agreeable conditions, i.e.,* $\alpha_j \leq \alpha_k$ *implies* $d_j \leq d_k$ *for all the jobs* J_j *and* J_k*, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* d_j *(i.e., the earliest due date (EDD) rule).*

Proof: Consider an optimal schedule π that does not follow the EDD rule. In this schedule there must be at least two adjacent jobs, say J_i and J_j, such that $d_i > d_j$, it implies $\alpha_i \geq \alpha_j$. Schedule π' is obtained from schedule π by interchanging jobs in the rth and in the $(r+1)$th positions of π. From the proof of Theorem 3, under π, the lateness of the jobs are

$$L_i(\pi) = (t + \frac{b}{c})(1 + c\alpha_i r^a) - \frac{b}{c} - d_i,$$

$$L_j(\pi) = (t + \frac{b}{c})(1 + c\alpha_i r^a)(1 + c\alpha_j (r+1)^a) - \frac{b}{c} - d_j,$$

whereas under π', they are

$$L_j(\pi') = (t + \frac{b}{c})(1 + c\alpha_j r^a) - \frac{b}{c} - d_j,$$

$$L_i(\pi') = (t + \frac{b}{c})(1 + c\alpha_j r^a)(1 + c\alpha_i (r+1)^a) - \frac{b}{c} - d_i.$$

Since $d_i > d_j$ and $\alpha_i \geq \alpha_j$, then

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

Hence, interchanging the positions of the jobs J_j and J_k will not increase the value of $L_{\max}$. A finite number of such changes transform π into the EDD order, showing that EDD sequence is optimal. □

Using the simple job interchange technique, we can prove the following results. The problems $1|\alpha_j(b+ct)r^a, w_j = k\alpha_j|\sum w_j C_j$, $1|\alpha_j(b+ct)r^a, \alpha_j = \alpha|L_{\max}$ and $1|\alpha_j(b+ct)r^a, d_j = k\alpha_j|L_{\max}$ can be obtained by sequencing the jobs in non-decreasing order of α_j, d_j and d_j. An optimal solution for the problem $1|\alpha_j(b+ct)r^a, \alpha_j = \alpha|\sum w_j C_j$ can be obtained by sequencing the jobs in non-increasing order of w_j.

3 Flow Shop Problems

The flow shop scheduling problem is to schedule n jobs $J_1, J_2, \ldots, J_n$ on m machines $M_1, M_2, \ldots, M_m$. Job J_j consists of m operations $(O_{1j}, O_{2j}, \ldots, O_{mj})$. Operation O_{ij} has to be processed on machine $M_i, i = 1, 2, \ldots, m$. The processing of operation $O_{i+1,j}$ may start only after O_{ij} has been completed. A machine can handle one job at a time and preemption is not allowed. All the jobs are available for processing at some time $t_0 \geq 0$. We also restrict ourselves to permutation schedules only. Let $p_{i,j,r}(t)$ be the processing time of job J_j on machine M_i if it is started at time t and scheduled in position r in a sequence. As in Section 2, we consider the following model,

$$p_{i,j,r}(t) = \alpha_{ij}(b + ct)r^a. \tag{5}$$

For a given schedule π, $C_{ij} = C_{ij}(\pi)$ represents the completion time of operation O_{ij}, and $C_j = C_{mj}$ represents the completion time of job J_j.

We first consider the special case of the two-machine flow shop problem where all the jobs have equal deterioration rates on machine M_2, i.e., $\alpha_{21} = \alpha_{22} = \ldots = \alpha_{2n} = \alpha_2$.

Theorem 5 *For the problem $F2|\alpha_{ij}(b + ct)r^a, \alpha_{2j} = \alpha_2|\sum C_j$, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of α_{1j}.*

Proof: Without loss of generality, let $\pi = [J_1, J_2, \ldots, J_n]$ be the schedule in which the jobs are processed in non-decreasing order of α_{1j}. Consider an arbitrary schedule $\pi' = [J_{[1]}, J_{[2]}, \ldots, J_{[n]}]$. We prove the theorem by showing that $C_j \leq C_{[j]}$ for $j = 1, 2, \ldots, n$.

The proof is by induction.

For $j = 1$, we have $C_1 = (t_0+\frac{b}{c})(1+c\alpha_{11})(1+c\alpha_2)-\frac{b}{c} \leq (t_0+\frac{b}{c})(1+c\alpha_{1[1]})(1+c\alpha_2)-\frac{b}{c} = C_{[1]}$.

Suppose that $C_j \leq C_{[j]}$ for $j = 1, 2, \ldots, k$, we have

$$C_{k+1} = (\max\{C_k, (t_0 + \frac{b}{c})\prod_{i=1}^{k+1}(1 + c\alpha_{1i}i^a) - \frac{b}{c}\} + \frac{b}{c})(1 + c\alpha_2(k + 1)^a) - \frac{b}{c},$$

$$C_{[k+1]} = (\max\{C_{[k]}, (t_0 + \frac{b}{c})\prod_{i=1}^{k+1}(1 + c\alpha_{1[i]}i^a) - \frac{b}{c}\} + \frac{b}{c})(1 + c\alpha_2(k + 1)^a) - \frac{b}{c}.$$

The term $(t_0 + \frac{b}{c})\prod_{i=1}^{k+1}(1 + c\alpha_{1i}i^a) - \frac{b}{c}$ is the makespan of the single machine problem with jobs $O_{11}, O_{12}, \ldots, O_{1,k+1}$ and the term $(t_0 + \frac{b}{c})\prod_{i=1}^{k+1}(1 + c\alpha_{1[i]}i^a) - \frac{b}{c}$ is the makespan of the single machine problem with jobs $O_{1[1]}, O_{1[2]}, \ldots, O_{1,[k+1]}$. Since $\alpha_{11} \leq \alpha_{12} \leq \ldots \leq \alpha_{1,k+1}$, by Theorem 1, we have

$$(t_0 + \frac{b}{c})\prod_{i=1}^{k+1}(1 + c\alpha_{1i}i^a) - \frac{b}{c} \leq (t_0 + \frac{b}{c})\prod_{i=1}^{k+1}(1 + c\alpha_{1[i]}i^a) - \frac{b}{c}.$$

From $(t_0 + \frac{b}{c})\prod_{i=1}^{k+1}(1 + c\alpha_{1i}i^a) - \frac{b}{c} \leq (t_0 + \frac{b}{c})\prod_{i=1}^{k+1}(1 + c\alpha_{1[i]}i^a) - \frac{b}{c}$ and $C_k \leq C_{[k]}$, we have $C_{k+1} \leq C_{[k+1]}$. Hence, the theorem holds for the job J_{k+1}, and from the induction principle, we have $C_n \leq C_{[n]}$. This completes the proof of the theorem. □

Theorem 6 *For the problem $F2|\alpha_{ij}(b + ct)r^a, \alpha_{2j} = \alpha_2|C_{\max}$, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of α_{1j}*

Proof: This result follows directly from the proof of Theorem 5. □

Now we consider a special case of the m-machine flow shop: The flow shop problem with dominant machines. Following Ho and Gupta [14], Xiang et al. [30] and Wang and Xia [31], machine M_r is dominated by M_k, or M_k dominates M_r iff $\max\{\alpha_{r,j}|j = 1, 2, \ldots, n\} \leq \min\{\alpha_{k,j}|j = 1, 2, \ldots, n\}$ (denoted $M_k > M_r$). Based upon the above concept of machine dominance, the case considered in the following is that the machines form an increasing series of dominating machines (idm), i.e.,

$$M_1 < M_2 < \ldots < M_m.$$

Theorem 7 *For the problem $Fm|\alpha_{ij}(b + ct)r^a, idm|\sum C_j$ and a fixed job in the first position, an optimal schedule can be obtained by sequencing the remaining $(n - 1)$ jobs in non-decreasing order of α_{mj}.*

Proof: For the schedule $\pi = [J_1, J_2, \ldots, J_n]$, since $M_1 < M_2 < \ldots < M_m$, similar to the result of Wang and Xia [34] (see also Figure 1), we have

$$C_1 = (t_0 + \frac{b}{c})\prod_{i=1}^{m}(1 + c\alpha_{i1}) - \frac{b}{c},$$

$$C_2 = (t_0 + \frac{b}{c})\prod_{i=1}^{m}(1 + c\alpha_{i1})(1 + c\alpha_{m2}2^a) - \frac{b}{c},$$

$$\ldots, \ldots$$

$$C_j = (t_0 + \frac{b}{c})\prod_{i=1}^{m}(1 + c\alpha_{i1})\prod_{k=2}^{j}(1 + c\alpha_{mk}k^a) - \frac{b}{c},$$

$$\ldots, \ldots$$

$$C_n = (t_0 + \frac{b}{c})\prod_{i=1}^{m}(1 + c\alpha_{i1})\prod_{k=2}^{n}(1 + c\alpha_{mk}k^a) - \frac{b}{c},$$

$$\sum C_j = (t_0 + \frac{b}{c})\prod_{i=1}^{m}(1 + c\alpha_{i1}) + (t_0 + \frac{b}{c})\prod_{i=1}^{m}(1 + c\alpha_{i1})\left[\sum_{j=2}^{n}\prod_{k=2}^{j}(1 + c\alpha_{mk}k^a)\right] - n\frac{b}{c}.$$

The term $(t_0+\frac{b}{c})\prod_{i=1}^{m}(1+c\alpha_{i1})$ is a constant (because job J_1 is fixed), and the term $\sum_{j=2}^{n}\prod_{k=2}^{j}(1+c\alpha_{mk}k^a$ can be minimized by sequencing the remaining $(n - 1)$ jobs in non-decreasing order of α_{mj} (by Theorem 2). Therefore, an optimal schedule for the $Fm|\alpha_{ij}(b + ct)r^a, idm|\sum C_j$ is obtained by arranging the remaining $(n - 1)$ jobs in non-decreasing order of α_{mj} provided that the first job is fixed. □

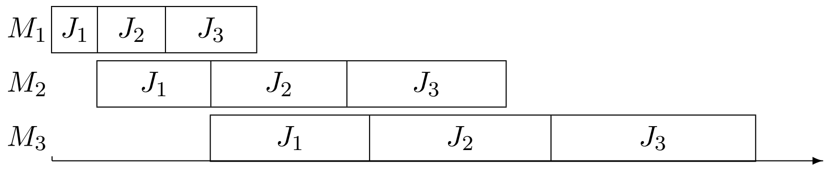

Figure 1. An Example of $Fm|\alpha_{ij}(b+ct)r^a, idm|f(C)$, $n=3, m=3$.

Theorem 8 *For the problem $Fm|\alpha_{ij}(b+ct)r^a, idm|C_{\max}$ and a fixed job in the first position, an optimal schedule can be obtained by sequencing the remaining $(n-1)$ jobs in non-decreasing order of α_{mj}.*

Proof: For the schedule $\pi = [J_1, J_2, \ldots, J_n]$, from the proof of Theorem 7, we have

$$C_{\max} = C_n = (t_0 + \frac{b}{c})\prod_{i=1}^{m}(1+c\alpha_{i1})\prod_{k=2}^{n}(1+c\alpha_{mk}k^a) - \frac{b}{c}.$$

The term $(t_0+\frac{b}{c})\prod_{i=1}^{m}(1+\alpha_{i1})$ is a constant, and the term $\prod_{k=2}^{n}(1+c\alpha_{mk}k^a)$ can be minimized by sequencing the remaining $(n-1)$ jobs in non-decreasing order of α_{mj} (by Theorem 1). □

To solve $Fm|\alpha_{ij}(b+ct)r^a, idm|\sum C_j$ and $Fm|\alpha_{ij}(b+ct)r^a, idm|C_{\max}$, each job can be considered in the first position to generate n schedules. The one with the minimum value of the performance measure among these n schedules is an optimal schedule.

Now, we consider another special case of the flow shop scheduling problem with identical deterioration rates on each machine, i.e., $\alpha_{ij} = \alpha_j$. For the classical problem $Fm|p_{ij} = p_j|f(C)$, where p_{ij} is the processing time of operation O_{ij}, the makespan is sequence independent (Pinedo [28]). Let $p_l = \max_{j=1,2,\ldots,n} p_j$, the makespan is

$$C_{\max}(\pi) = \sum_{j=1}^{n} p_j + (m-1)p_l.$$

The above solution can be generalized to the problem $Fm|\alpha_{ij}(b+ct)r^a, \alpha_{ij} = \alpha_j|f(C)$. We consider each operation as a job, and the problem $Fm|\alpha_{ij}(b+ct)r^a, \alpha_{ij} = \alpha_j|f(C)$ is equivalent to the single machine problem $1|\alpha_j(b+ct)r^a|f(C)$ with $n+m-1$ jobs. For the schedule $\pi = [J_1, J_2, \ldots, J_n]$, let the job J_l be considered as m jobs ($\alpha_l = \max\{\alpha_1, \alpha_2 2^a, \ldots, \alpha_n n^a\}$). Hence, the makespan of $Fm|\alpha_{ij}(b+ct)r^a, \alpha_{ij} = \alpha_j|f(C)$ is

$$C_{\max} = (t_0 + \frac{b}{c})(1+c\alpha_l)^{m-1}\prod_{i=1}^{n}(1+c\alpha_i i^a) - \frac{b}{c}.$$

By the results of the single machine problem, we have the following results:

Theorem 9 *For the problem* $Fm|\alpha_{ij}(b + ct)r^a, \alpha_{ij} = \alpha_j|C_{\max}$*, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* α_j *(the SDR rule).*

Theorem 10 *For the problem* $Fm|\alpha_{ij}(b + ct)r^a, \alpha_{ij} = \alpha_j|\sum C_j$*, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* α_j *(the SDR rule).*

Theorem 11 *For the problem* $Fm|\alpha_{ij}(b + ct)r^a, \alpha_{ij} = \alpha_j|\sum w_jC_j$*, if the jobs have agreeable weights, i.e.,* $\alpha_j \leq \alpha_k$ *implies* $w_j \geq w_k$ *for all the jobs* J_j *and* J_k*, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* α_j/w_j *(the WSDR rule).*

Theorem 12 *For the problem* $Fm|\alpha_{ij}(b + ct)r^a, \alpha_{ij} = \alpha|\sum w_jC_j$*, an optimal schedule can be obtained by sequencing the jobs in non-increasing order of* w_j*.*

Theorem 13 *For the problem* $Fm|\alpha_{ij}(b+ct)r^a, \alpha_{ij} = \alpha_j, w_j = k\alpha_j|\sum w_jC_j$*, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* α_j *(the SDR rule).*

Theorem 14 *For the problem* $Fm|\alpha_{ij}(b + ct)r^a, \alpha_{ij} = \alpha_j|L_{\max}$*, if the jobs have agreeable conditions, i.e.,* $\alpha_j \leq \alpha_k$ *implies* $d_j \leq d_k$ *for all the jobs* J_j *and* J_k*, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* d_j *(the EDD rule).*

Theorem 15 *For the problem* $Fm|\alpha_{ij}(b + ct)r^a, \alpha_{ij} = \alpha|L_{\max}$*, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* d_j *(the EDD rule).*

Theorem 16 *For the problem* $Fm|\alpha_{ij}(b + ct)r^a, \alpha_{ij} = \alpha_j, d_j = k\alpha_j|L_{\max}$*, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* d_j *(the EDD rule).*

4 Conclusions

Different types of scheduling problems with the effects of deterioration and learning were studied in this paper. It was shown that for some special cases of the single machine problem and flow shop problem, they can be solved in polynomial time (see Table 1). However, the complexity status of the problems of minimizing the total weighted completion time and maximum lateness is still open (see Table 1). These questions may be a subject for a future research.

As a side result of our analysis, it is easily shown that some bicriterion problems (simultaneous optimization) can be solved. (For example, $F2|\alpha_{ij}(b + ct)r^a, \alpha_{2j} = \alpha_2|C_{\max} \cap \sum C_j$, $Fm|\alpha_{ij}(b + ct)r^a, idm|C_{\max} \cap \sum C_j$).

Table 1

Problem	Complexity
$1\|\alpha_j(b+ct)r^a\|C_{\max}$	$O(n\log n)$
$1\|\alpha_j(b+ct)r^a\|\sum C_j$	$O(n\log n)$
$1\|\alpha_j(b+ct)r^a, \alpha_j \leq \alpha_k \Longrightarrow w_j \geq w_k\|\sum w_jC_j$	$O(n\log n)$
$1\|\alpha_j(b+ct)r^a, w_j = k\alpha_j\|\sum w_jC_j$	$O(n\log n)$
$1\|\alpha_j(b+ct)r^a, \alpha_j = \alpha\|\sum w_jC_j$	$O(n\log n)$
$1\|\alpha_j(b+ct)r^a, \alpha_j \leq \alpha_k \Longrightarrow d_j \leq d_k\|L_{\max}$	$O(n\log n)$
$1\|\alpha_j(b+ct)r^a, d_j = k\alpha_j\|L_{\max}$	$O(n\log n)$
$1\|\alpha_j(b+ct)r^a, \alpha_j = \alpha\|L_{\max}$	$O(n\log n)$
$F2\|\alpha_{ij}(b+ct)r^a, \alpha_{2j} = \alpha_2\|C_{\max}$	$O(n\log n)$
$F2\|\alpha_{ij}(b+ct)r^a, \alpha_{2j} = \alpha_2\|\sum C_j$	$O(n\log n)$
$Fm\|\alpha_{ij}(b+ct)r^a, idm\|C_{\max}$	$O(n^2\log n)$
$Fm\|\alpha_{ij}(b+ct)r^a, idm\|\sum C_j$	$O(n^2\log n)$
$Fm\|\alpha_{ij}(b+ct)r^a, \alpha_{ij} = \alpha_j\|C_{\max}$	$O(n\log n)$
$Fm\|\alpha_{ij}(b+ct)r^a, \alpha_{ij} = \alpha_j\|\sum C_j$	$O(n\log n)$
$Fm\|\alpha_{ij}(b+ct)r^a, \alpha_{ij} = \alpha_j, \alpha_j \leq \alpha_k \Longrightarrow w_j \geq w_k\|\sum w_jC_j$	$O(n\log n)$
$Fm\|\alpha_{ij}(b+ct)r^a, \alpha_{ij} = \alpha_j, w_j = k\alpha_j\|\sum w_jC_j$	$O(n\log n)$
$Fm\|\alpha_{ij}(b+ct)r^a, \alpha_{ij} = \alpha_j, \alpha_j = \alpha\|\sum w_jC_j$	$O(n\log n)$
$Fm\|\alpha_{ij}(b+ct)r^a, \alpha_{ij} = \alpha_j, \alpha_j \leq \alpha_k \Longrightarrow d_j \leq d_k\|L_{\max}$	$O(n\log n)$
$Fm\|\alpha_{ij}(b+ct)r^a, \alpha_{ij} = \alpha_j, d_j = k\alpha_j\|L_{\max}$	$O(n\log n)$
$Fm\|\alpha_{ij}(b+ct)r^a, \alpha_{ij} = \alpha_j, \alpha_j = \alpha\|L_{\max}$	$O(n\log n)$
$1\|\alpha_j(b+ct)r^a\|\sum w_jC_j$	Open problem
$1\|\alpha_j(b+ct)r^a\|L_{\max}$	Open problem
$F2\|\alpha_{ij}(b+ct)r^a\|C_{\max}$	Open problem
$Fm\|\alpha_{ij}(b+ct)r^a, \alpha_{ij} = \alpha_j, \|\sum w_jC_j$	Open problem
$Fm\|\alpha_{ij}(b+ct)r^a, \alpha_{ij} = \alpha_j, \|L_{\max}$	Open problem

Acknowledgements— We are grateful to 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 China Business Services*. The research of the first author was also partially supposed by the foundation of Shenyang Institute of Aeronautical Engineering under grant number: 05YB08.

References

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

[2] A. Bachman T.C.E. Cheng, A. Janiak and C.T. Ng (2002). Scheduling start time dependent jobs to minimize the total weighted completion time. Journal of the Operational Research Society, 53, 688-693.

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

[4] A. Bachman and A. Janiak (2004). Scheduling jobs with position-dependent processing times. Journal of the Operational Research Society, 55, 257-264.

[5] A. Bachman, A. Janiak and M.Y. Kovalyov (2002). Minimizing the total weighted completion time of deteriorating jobs. Information Processing Letters, 81 (2), 81-84.

[6] A.B. Badiru (1992). Computational survey of univariate and multivariate learning curve models. IEEE transactions on Engineering Management, 39, 176-188.

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

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

[9] Z-L. Chen (1996). Parallel machine scheduling with time dependent processing times. Discrete Applied Mathematics, 70, 81-94.

[10] T.C.E. Cheng and Q. Ding (1998). The complexity of scheduling starting time dependent task with release dates. Information Processing Letters, 65, 75-79.

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

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

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

[14] J.C. Ho and J.N.D. Gupta (1995). Flowshop scheduling with dominant machines. Computers and Operations Research, 22, 237-246.

[15] K.I.-J. Ho, J.Y.-T. Leung and W.-D. Wei (1993). Complexity of scheduling tasks with time-dependent execution times. Information Processing Letters, 48, 315-320.

[16] A.S. Kunnathur and S. K. Gupta (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.

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

[18] G. Mosheiov (1991). V-shaped policies for scheduling deteriorating jobs. Operations Research, 39, 979-991.

[19] G. Mosheiov (1994). Scheduling jobs under simple linear deterioration. Computers and Operations Research, 21, 653-659.

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

[21] G. Mosheiov (1998). Multi-machine scheduling with linear deterioration. INFOR, 36, 205-214.

[22] G. Mosheiov (2002). Complexity analysis of job-shop scheduling with deteriorating jobs. Discrete Applied Mathematics, 117, 195-209.

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

[24] G. Mosheiov (2001). Parallel machine scheduling with a learning effect. Journal of the Operational Research Society, 52, 1165-1169.

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

[26] D.A Nembhard and N. Osothsilp (2002). Task complexity effects on between-individual learning/forgetting variability. International Journal of Industrial Ergonomics, 29, 297C306.

[27] C.T. Ng, T.C.E. Cheng, A. Bachman, and A. Janiak, (2002). Three scheduling problems with deteriorating jobs to minimize the total completion time. Information Processing Letters, 81, 327-333.

[28] M. Pinedo, (2002), Scheduling: Theory, Algorithms, and Systems, Upper Saddle River, NJ: Prentice-Hall.

[29] P.S. Sundararaghavan and A.S. Kunnathur (1994). Single machine scheduling with start time dependent processing times: some solvable cases. European Journal of Operational Research, 78, 394-403.

[30] S. Xiang, G. Tang and T.C.E. Cheng (2000). Solvable cases of permutation flowshop scheduling with dominating machines. International Journal of Production Economics, 66, 53-57.

[31] J.B. Wang and Z.Q. Xia (2005). No-wait or no-idle permutation flowshop scheduling with dominating machines. Journal of Applied Mathematics and Computing, 17, 419-432.

[32] J.B. Wang and Z.Q. Xia (2005). Scheduling jobs under decreasing linear deterioration. Information Processing Letters, 94, 63-69.

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

[34] J.B. Wang and Z.Q. Xia. Flow shop scheduling with deteriorating jobs under dominating machines. Omega (in press).

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