Single machine scheduling with a time-dependent learning effect

Ji-Bo Wang,[a,b,1] C.T. Ng,[b] 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

In this paper we consider the single machine scheduling problem with a time-dependent learning effect. The time-dependent learning effect of a job is assumed to be a function of the total normal processing time of the jobs scheduled in front of the job. For the following three objective functions: the weighted sum of completion times, the maximum lateness and the number of tardy jobs, we show by examples that the optimal schedule of the classical version is not optimal in the presence of a time-dependent learning effect. But for some special cases, we prove that the weighted shortest processing time (WSPT) rule, the earliest due date (EDD) rule and Moore's algorithm can construct an optimal sequence for these objective functions, respectively. We also give the worst-case error bound of these three rules for the general cases.

Keywords: Scheduling, Single machine, Learning effect, Time-dependent

1 Introduction

In classical scheduling problems the processing time of a job is assumed to be a constant. However, in many realistic problems of operations management, both machines and workers can improve their performance by repeating the production operations. Therefore, the actual processing time of a job is shorter if it is scheduled later in a sequence. This phenomenon is known as the "learning effect" in the literature (Badiru [2]). Although learning theory was first applied to industry more than 60 years ago (Wright [20]), it appears to have become a topic in scheduling research only in recent years.

Biskup [3] and Cheng and Wang [5] 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 [20]. Biskup [3] proved that single machine scheduling with a learning effect remains polynomial solvable if its objective is to minimize the deviation of job completion times from a common due date or to minimize the sum of job flow times. Cheng and Wang [5] considered a single machine scheduling problem in which the job processing times decrease as a result of learning. A volume-dependent piecewise linear processing time function

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

was used to model the learning effect. The objective is to minimize the maximum lateness. They showed that the problem is NP-hard in the strong sense and then identified two special cases that are polynomially solvable. They also proposed two heuristics and analysed their worst-case performance. Mosheiov [13, 14] investigated several other single machine problems and the minimum total flow time problem on identical parallel machines. Liu et al. [11] proved that the weighted shortest processing time (WSPT) rule, the earliest due date (EDD) rule and the modified Moore-Hodgson algorithm can construct optimal sequences under certain conditions for the following three objectives: the total weighted completion time, the maximum lateness and the number of tardy jobs, respectively. They also gave an error estimation for each of these rules for the general cases. Lee and Wu [9] proposed a heuristic algorithm to solve the total completion time minimization problem in a two-machine flow shop with a learning effect. Lee et al. [10] studied the learning effect in a bi-criterion single machine scheduling problem, with the objective of minimizing a linear combination of the total completion time and the maximum tardiness. They presented a branch-and-bound and a heuristic algorithm to search for optimal and near optimal solutions. Mosheiov and Sidney [15] considered a job-dependent learning curve, where the learning rate of some jobs is faster than that of the others. Biskup and Simons [4] considered a scheduling problem where the processing times decrease according to a learning rate, which can be influenced by an initial cost-incurring investment. They presented a formulation of the common due date scheduling problem with autonomous and induced learning effects. They further proved some structural properties, which enable the development of a polynomial bound solution procedure. Mosheiov and Sidney [16] introduced a polynomial time solution for the single machine scheduling problem to minimize the number of tardy jobs with general non-increasing job-dependent learning curves and a common due-date. Wang [18] considered flow shop scheduling problems with a learning effect. He suggested the use of Johnson's rule as a heuristic algorithm for two-machine flow shop scheduling to minimize the makespan. He also developed polynomial time solution algorithms for some special cases of the following objective functions: the weighted sum of completion times and the maximum lateness. Wang and Xia [19] considered the same problem of Wang [18]. The objective was to minimize one of two regular performance measures, namely the makespan and the 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 for two special cases of the problem, i.e., identical processing times on each machine and an increasing series of dominating machines. Kuo and Yang [7] considered a single machine scheduling problem with a time-dependent learning effect. The time-dependent learning effect of a job is assumed to be a function of the total normal processing time of the jobs scheduled in front of it. They showed that the SPT-sequence is the optimal sequence for the objective of minimizing the total completion time. Kuo and Yang [8] considered the single machine group scheduling problem with a time-dependent learning effect. They showed that the single machine group scheduling problem with a time-dependent learning effect remains polynomially solvable for the objectives of minimizing the makespan and minimizing the total completion time. A survey of this line of scheduling research can be found in Bachman and Janiak [1].

In this paper we consider the same model as that of Kuo and Yang [7], but with different objective functions. The remaining part of this paper is organized as follows. In Section 2 we formulate the model. In Section 3 we consider three classical single machine scheduling problems. The last section presents the conclusions.

2 Problem formulation

The focus of this paper is to study the time-dependent learning effect in scheduling. The model is described as follows. There are given a single machine and n independent and non-preemptive jobs that are immediately available for processing. The machine can handle one job at a time and preemption is not allowed. Let p_j be the normal processing time of job j and $p_{[k]}$ be the normal processing time of a job if it is scheduled in the kth position in a sequence. Associated with each job j $(j = 1, 2, ..., n)$ is a weight w_j and a due-date d_j. Let $p_{j,r}$ be the processing time of job j if it is scheduled in position r in a sequence. Then

$$p_{j,r} = p_j(1 + p_{[1]} + p_{[2]} + \ldots + p_{[r-1]})^a, \tag{1}$$

where $a \leq 0$ is a constant learning index. It is clear that the learning effect satisfies the following conditions: $0 < (1 + \sum_{k=1}^{r-1} p_{[k]})^a \leq 1$.

For a given schedule $\pi = [[1], [2], \ldots, [n]]$, $C_j = C_j(\pi)$ represents the completion time of job j and $f(C) = f(C_1, C_2, \ldots, C_n)$ is a regular measure of performance. Let $\sum w_j C_j$, $L_{\max} = \max\{C_j - d_j | j = 1, 2, \ldots, n\}$ and $\sum U_j$, where $U_j = 1$ if $C_j > d_j$ (i.e., the job is late) and $U_j = 0$ otherwise, $j = 1, 2, \ldots, n$, represent the weighted sum of completion times, the maximum lateness and the number of tardy jobs of a given permutation, respectively. For convenience, we denote the time-dependent learning effect given by (1) by LE_t [7]. We use $1|LE_t|f(C)$ to denote single machine scheduling with a time-dependent learning effect that is job-independent.

3 Several single machine scheduling problems

First, we give several lemmas, which are useful for the following theorems.

Lemma 1 *$(x - y)b^a + y(b + x)^a - x(b + y)^a \leq 0$ if $0 < x \leq y$, $b \geq 1$ and $a \leq 0$.*

Proof The proof can be found in Kuo and Yang [7]. □

Lemma 2 *[8] For the problem $1|LE_t|C_{\max}$, there exists an optimal schedule in which the job sequence is determined by the SPT rule.*

Lemma 3 *If $p_{i_1} \leq p_{i_2} \leq \ldots \leq p_{i_m} \leq p_{i_{m+1}}$, then the makespan of sequence $(i_1, i_2, \ldots, i_m)$ is not greater than the makespan of any sequence of the form*

$$(i_1, i_2, \ldots, i_{k-1}, i_{k+1}, \ldots, i_{m+1}), k \in \{1, 2, \ldots, m\}.$$

Proof The proof is trivial and omitted. □

Similar to Mosheiov [13], for the next three problems, we show that the optimal schedule of the classical version is not optimal for the problems $1|LE_t|\sum w_jC_j$, $1|LE_t|L_{\max}$ and $1|LE_t|\sum U_j$, respectively.

Example 1. $n = 2, p_1 = 1, p_2 = 2, w_1 = 10, w_2 = 21, a = -0.5$. The schedule according to the WSPT rule is $[2, 1]$, yielding the value $\sum w_jC_j = 67.77$. Obviously, the optimal sequence is $[1, 2]$, yielding the optimal value $\sum w_jC_j = 60.70$.

In order to solve the problem approximately, we will use the WSPT rule as a heuristic for the problem $1|LE_t|\sum w_jC_j$. The performance of the heuristic will be evaluated by its worst-case error bound.

Theorem 1 *Let S^* be an optimal schedule and S be a WSPT schedule for the problem $1|LE_t|\sum w_jC_j$. Then $\sum w_jC_j(S)/\sum w_jC_j(S^*) \leq 1/(1+\sum_{j=1}^{n} p_j - p_{\min})^a$, where $p_{\min} = \min\{p_j|j = 1, 2, \ldots, n\}$, and the bound is tight.*

Proof Without loss of generality, we can suppose that $p_1/w_1 \leq p_2/w_2 \leq \ldots \leq p_n/w_n$. Then

$$\begin{aligned}
\sum w_jC_j(S) &= w_1p_1 + w_2[p_1 + p_2(1 + p_1)^a] + \ldots + w_n[p_1 + p_2(1 + p_1)^a \\
&\quad + \ldots + p_n(1 + p_1 + p_2 + \ldots + p_{n-1})^a] \\
&\leq \sum_{j=1}^{n} w_j(\sum_{k=1}^{j} p_k),
\end{aligned}$$

$$\begin{aligned}
\sum w_jC_j(S^*) &= w_{[1]}p_{[1]} + w_{[2]}[p_{[1]} + p_{[2]}(1 + p_{[1]})^a] + \ldots + w_{[n]}[p_{[1]} + p_{[2]}(1 + p_{[1]})^a \\
&\quad + \ldots + p_{[n]}(1 + p_{[1]} + p_{[2]} + \ldots + p_{[n-1]})^a] \\
&\geq \sum_{j=1}^{n} w_{[j]}(\sum_{k=1}^{j} p_{[k]})(1 + \sum_{j=1}^{n-1} p_{[j]})^a \\
&\geq \sum_{j=1}^{n} w_j(\sum_{k=1}^{j} p_k)(1 + \sum_{j=1}^{n} p_j - p_{\min})^a,
\end{aligned}$$

hence

$$\sum w_j C_j(S)/\sum w_j C_j(S^*) \leq 1/(1+\sum_{j=1}^{n} p_j - p_{\min})^a.$$

It is not difficult to see that the bound is tight, since if a=0, we have $\sum w_j C_j(S)/\sum w_j C_j(S^*) = 1$. This result is intuitive as when a=0, the WSPT schedule is optimal. □

Obviously, the result obtained $\sum w_j C_j(S)/\sum w_j C_j(S^*)$ depends greatly on the parameter values.

Example 2. $n = 2, p_1 = 1, p_2 = 100, d_1 = 1, d_2 = 0, a = -0.5$. The schedule according to the EDD rule is $[2,1]$, yielding the value $L_{\max} = 100$. Obviously, the optimal sequence is $[1,2]$, yielding the optimal value $L_{\max} = 71.7$.

In order to solve the problem approximately, we will use the EDD rule as a heuristic for the problem $1|LE_t|L_{\max}$. To develop a worst-case performance ratio for a heuristic, we have to avoid cases involving nonpositive $L_{\max}$. Similar to Cheng and Wang [5], the worst-case error bound is defined as follows:

$$\frac{L_{\max}(S)+d_{\max}}{L_{\max}(S^*)+d_{\max}},$$

where S and $L_{\max}(S)$ denote the heuristic schedule and the corresponding maximum lateness, respectively, while S^* and $L_{\max}(S^*)$ denote the optimal schedule and the minimum maximum lateness value, respectively, and $d_{\max} = \max\{d_j | j = 1, 2, \ldots, n\}$.

Theorem 2 *Let S^* be an optimal schedule and S be an EDD schedule for the problem $1|LE_t|L_{\max}$. Then*

$$\frac{L_{\max}(S)+d_{\max}}{L_{\max}(S^*)+d_{\max}} \leq \frac{\sum_{i=1}^{n} p_i}{C^*_{\max}},$$

and the bound is tight, where $C^*_{\max}$ is the optimal makespan of the problem $1|LE_t|C_{\max}$.

Proof Without loss of generality, we can suppose that $d_1 \leq d_2 \leq \ldots \leq d_n$, then

$$\begin{aligned} L_{\max}(S) &= \max\{p_1 + p_2(1+p_1)^a + \ldots + p_j(1+p_1+p_2+\ldots+p_{j-1})^a \\ &\quad -d_j | j = 1, 2, \ldots, n\} \\ &\leq \max\{p_1 + p_2 + \ldots + p_j - d_j | j = 1, 2, \ldots, n\} \\ &= L'_{\max}(S), \end{aligned}$$

where $L'_{\max}(S)$ is the optimal value of the classical version of the problem, i.e., $p_{j,r} = p_j$.

$$L_{\max}(S^*) = \max\{p_{[1]} + p_{[2]}(1+p_{[1]})^a + \ldots + p_{[j]}(1+p_{[1]}+p_{[2]}+\ldots+p_{[j-1]})^a$$

$$
\begin{aligned}
& -d_{[j]}|j = 1, 2, \ldots, n\} \\
= \;& \max\{\sum_{i=1}^{j} p_{[i]} - d_{[j]} - \sum_{i=1}^{j} p_{[i]} + \sum_{i=1}^{j} p_{[i]}(1 + \sum_{k=1}^{i-1} p_{[k]})^a | j = 1, 2, \ldots, n\} \\
\geq \;& \max\{\sum_{i=1}^{j} p_{[i]} - d_{[j]} | j = 1, 2, \ldots, n\} - \sum_{i=1}^{n} p_{[i]} + \sum_{i=1}^{n} p_{[i]}(1 + \sum_{k=1}^{i-1} p_{[k]})^a \\
\geq \;& L'_{\max}(S) - \sum_{i=1}^{n} p_i + C^*_{\max},
\end{aligned}
$$

hence,

$$L_{\max}(S) - L_{\max}(S^*) \leq \sum_{i=1}^{n} p_i - C^*_{\max},$$

and so

$$\frac{L_{\max}(S) + d_{\max}}{L_{\max}(S^*) + d_{\max}} \leq 1 + \frac{\sum_{i=1}^{n} p_i - C^*_{\max}}{L_{\max}(S^*) + d_{\max}} \leq 1 + \frac{\sum_{i=1}^{n} p_i - C^*_{\max}}{C^*_{\max}} \leq \frac{\sum_{i=1}^{n} p_i}{C^*_{\max}},$$

where $C^*_{\max}$ can be obtained by the SPT rule (see Lemma 2).

It is not difficult to see that the bound is tight, since if a=0, we have $C_{\max} = \sum_{i=1}^{n} p_i$ and $\frac{L_{\max}(S)+d_{\max}}{L_{\max}(S^*)+d_{\max}} = 1$. This result is intuitive as when a=0, the EDD schedule is optimal. □

Let J denote the set of jobs already scheduled, J^d be the set of jobs already considered for scheduling, but that have been discarded as they will not meet their due dates in the optimal schedule, and J^c denote the set of jobs not yet considered for scheduling. The problem $1||\sum U_j$ is known to be solved by Moore's algorithm [12] as follows:

Moore's Algorithm.

Step 1: Order the jobs in non-decreasing order of their due dates (EDD).
Step 2: If no jobs in the sequence are late, stop. The schedule is optimal.
Step 3: Find the first late job in the schedule. Denote this job by α.
Step 4: Find a job β with $p_\beta = \max_{i=1,2,\ldots,\alpha} p_i$. Remove job β from the schedule and process it after the completion of all the jobs that were processed. Go to Step 2.

As a special case, it is known by Jackson's lemma [6] that if a schedule with no tardy jobs exists, then the EDD sequence will contain no tardy jobs. Example 3 shows that Jackson's lemma does not hold for $1|LE_t|\sum U_j$ (therefore, Moore's Algorithm is not optimal for the problem).

Example 3. $n = 2, p_1 = 1, p_2 = 100, d_1 = 91, d_2 = 90, a = -0.5$. The schedule according to the EDD rule is $[2, 1]$, yielding the value $\sum U_j = 2$. Obviously, the optimal sequence is $[1, 2]$, yielding the optimal value $\sum U_j = 0$.

In order to solve the problem approximately, we will use Moore's Algorithm as a heuristic for the problem $1|LE_t|\sum U_j$. The performance of the heuristic will be evaluated by its worst-case error bound.

Theorem 3 *Let S^* be an optimal schedule and S be the schedule obtained by Moore's Algorithm for the problem $1|LE_t|\sum U_j$. Then*

$$\sum U_j(S) - \sum U_j(S^*) \leq n - 1.$$

Proof It suffices to prove that $\sum U_j(S) \leq n-1$ for the case $\sum U_j(S^*) = 0$. Without loss of generality, we can suppose that $d_1 \leq d_2 \leq \ldots \leq d_n$, the optimal sequence is $[i_1, i_2, \ldots, i_n]$, and in this optimal sequence job 1 is the m-th job processed, $m \geq 1$. Then by $\sum U_j(S^*) = 0$, we know that

$$p_{i_1} + p_{i_2}(1 + p_{i_1})^a + \ldots + p_{i_m}(1 + p_{i_1} + p_{i_2} + \ldots + p_{i_{m-1}})^a \leq d_{i_m},$$

in which $i_m = 1$, so $p_{i_1} \leq d_{i_m} \leq d_{i_1}$.

Thus, as we solve the problem by Moore's Algorithm, if there is a job completed in time in J as we pick the jobs successively from $J^c = (1, 2, \ldots, n)$ and put them into set J, then the theorem has been proved; if all the job scheduled before job i_1 in J^c are tardy in J as they are picked successively from J^c and put into J, then by Moore's Algorithm the next job that may be picked from J^c is job i_1. By $p_{i_1} \leq d_{i_1}$, we know that job i_1 is completed on time, thus $\sum U_j(S) \leq n - 1$. The proof is completed. □

For the three objective functions of minimizing the weighted sum of completion times, minimizing the maximum lateness, and minimizing the number of tardy jobs, the above examples show that the optimal solutions for the classical versions do not hold with a time-dependent learning effect. But some special cases of these scheduling problems with a time-dependent learning effect modelled as (1) can be solved in polynomial time.

Theorem 4 *For the problem $1|LE_t|\sum w_j C_j$, if the jobs have agreeable weights, i.e., $p_j \leq p_k$ implies $w_j \geq w_k$ for all the jobs j and k, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of p_j/w_j, i.e., the WSPT rule is optimal.*

Proof. (By contradiction). Consider an optimal schedule π that does not follow the WSPT rule. In this schedule there must be at least two adjacent jobs, say job j followed by job k, such that $p_j/w_j > p_k/w_k$, which implies $p_j \geq p_k$. Assume that job j is scheduled in position r. Perform an adjacent pair-wise interchange of jobs j and k. Whereas under the original schedule π job j is scheduled in position r and job k is scheduled in position $r+1$, under the new schedule job k is scheduled in position r and job j 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 and k are not affected by the interchange. Furthermore, the completion times of the jobs processed after jobs j and k cannot be increased by the interchange since $p_j \geq p_k$. Under π,

$$C_j(\pi) = p_{[1]} + p_{[2]}(1 + p_{[1]}) + \ldots + p_{[r-1]}(1 + \sum_{q=1}^{r-2} p_{[q]})^a + p_j(1 + \sum_{q=1}^{r-1} p_{[q]})^a,$$

$$C_k(\pi) = p_{[1]} + p_{[2]}(1 + p_{[1]}) + \ldots + p_{[r-1]}(1 + \sum_{q=1}^{r-2} p_{[q]})^a + p_j(1 + \sum_{q=1}^{r-1} p_{[q]})^a + p_k(1 + \sum_{q=1}^{r-1} p_{[q]} + p_j)^a,$$

whereas under π', they are

$$C_k(\pi') = p_{[1]} + p_{[2]}(1 + p_{[1]}) + \ldots + p_{[r-1]}(1 + \sum_{q=1}^{r-2} p_{[q]})^a + p_k(1 + \sum_{q=1}^{r-1} p_{[q]})^a,$$

$$C_j(\pi') = p_{[1]} + p_{[2]}(1 + p_{[1]}) + \ldots + p_{[r-1]}(1 + \sum_{q=1}^{r-2} p_{[q]})^a + p_k(1 + \sum_{q=1}^{r-1} p_{[q]})^a + p_j(1 + \sum_{q=1}^{r-1} p_{[q]} + p_k)^a.$$

So we have

$$\begin{aligned}
& \sum w_j C_j(\pi) - \sum w_j C_j(\pi') \\
\geq \;& w_j p_j(1 + \sum_{q=1}^{r-1} p_{[q]})^a + w_k[p_j(1 + \sum_{q=1}^{r-1} p_{[q]})^a + p_k(1 + \sum_{q=1}^{r-1} p_{[q]} + p_j)^a] \\
& - w_k p_k(1 + \sum_{q=1}^{r-1} p_{[q]})^a - w_j[p_k(1 + \sum_{q=1}^{r-1} p_{[q]})^a + p_j(1 + \sum_{q=1}^{r-1} p_{[q]} + p_k)^a] \\
= \;& (w_j + w_k)\left((p_j - p_k)(1 + \sum_{q=1}^{r-1} p_{[q]})^a + p_k(1 + \sum_{q=1}^{r-1} p_{[q]} + p_j)^a - p_j(1 + \sum_{q=1}^{r-1} p_{[q]} + p_k)^a\right) \\
& + w_k p_j(1 + \sum_{q=1}^{r-1} p_{[q]} + p_k)^a) - w_j p_k(1 + \sum_{q=1}^{r-1} p_{[q]} + p_j)^a.
\end{aligned}$$

Since $p_j \geq p_k$, $w_k p_j > w_j p_k$ (because $p_j/w_j > p_k/w_k$), $(1+\sum_{q=1}^{r-1} p_{[q]}+p_k)^a \geq (1+\sum_{q=1}^{r-1} p_{[q]}+p_j)^a$ (because x^a is non-increasing for $a \leq 0$) and $(p_j - p_k)(1 + \sum_{q=1}^{r-1} p_{[q]})^a + p_k(1 + \sum_{q=1}^{r-1} p_{[q]} + p_j)^a - p_j(1 + \sum_{q=1}^{r-1} p_{[q]} + p_k)^a \geq 0$ (due to Lemma 1), then $\sum w_j C_j(\pi) - \sum w_j C_j(\pi') > 0$. It follows that the weighted sum of completion times under π' is strictly less than that under π. This contradicts the optimality of π and proves the theorem. □

The following two theorems are corollaries of Theorem 4.

Theorem 5 *For the problem $1|LE_t, p_j = p|\sum w_j C_j$, an optimal schedule can be obtained by sequencing the jobs in non-increasing order of w_j.*

Theorem 6 *For the problem* $1|LE_t, w_j = kp_j|\sum w_jC_j$*, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* p_j*, i.e., the SPT rule is optimal.*

Theorem 7 *For the problem* $1|LE_t|L_{\max}$*, if the jobs have agreeable conditions, i.e.,* $p_i \leq p_j$ *implies* $d_i \leq d_j$ *for all the jobs* i *and* j*, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* d_j*, i.e., the EDD rule is optimal.*

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 and k in the rth and (r+1)th positions of π, respectively, such that $d_j > d_k$, which implies $p_j \geq p_k$. 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 4, under π, the lateness of the jobs are

$$L_j(\pi) = C_j(\pi) - d_j,$$

$$L_k(\pi) = C_k(\pi) - d_k,$$

whereas under π', they are

$$L_k(\pi') = C_k(\pi') - d_k,$$

$$L_j(\pi') = C_j(\pi') - d_j.$$

Since $d_j > d_k$ and $p_j \geq p_k$, we have $C_j(\pi') \leq C_k(\pi)$ and $C_k(\pi') \leq C_j(\pi)$ [8], hence it is easily verified that

$$\max\{L_j(\pi'), L_k(\pi')\} < \max\{L_j(\pi), L_k(\pi)\}.$$

Hence, interchanging the positions of jobs j and k will decrease the value of $L_{\max}$. This is a contradiction. □

Theorem 8 *For the problem* $1|LE_t, p_j = p|L_{\max}$*, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* d_j*, i.e., the EDD rule is optimal.*

If $d_j = kp_j$, the jobs have agreeable conditions, i.e., $p_i \leq p_j$ implies $d_i \leq d_j$ for all the jobs i and j. Hence, the following corollary can be easily obtained.

Corollary 1 *For the problem* $1|LE_t, d_j = kp_j|L_{\max}$*, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* d_j*, i.e., the EDD rule is optimal.*

Theorem 9 *For the problem* $1|LE_t|\sum U_j$*, if the jobs have agreeable conditions, i.e.,* $p_i \leq p_j$ *implies* $d_i \leq d_j$ *for all the jobs* i *and* j*, an optimal schedule can be obtained by Moore's Algorithm.*

Proof. Without loss of generality, we can assume that $d_1 \leq d_2 \leq \ldots \leq d_n$, then under the condition of the theorem, we can further assume $p_1 \leq p_2 \leq \ldots \leq p_n$. Let J_k denote a set of jobs $\{1, 2, \ldots, k\}$ that satisfies the following two conditions :

(a) All the jobs in J_k have the maximum number of no-late jobs, say n_k.

(b) Among all sets with n_k no-late jobs among the first k jobs, J_k is a set such that it has the smallest total processing time.

Note that set J_n corresponds to an optimal schedule. As in Pinedo [17], the proof that the algorithm leads to J_n is by induction.

Obviously, for $k = 1$, it is true that the algorithm that constructs J_1 satisfies the conditions (a) and (b).

Suppose for $k = m$, the set $J_m = \{1, 2, \ldots, m\}$ constructed by Moore's Algorithm satisfies the conditions (a) and (b). Now we prove for $k = m+1$, the set J_{m+1} constructed starting with set J_m also satisfies the conditions (a) and (b). There are two cases to be considered.

Case 1. Job $m + 1$ is completed by its due date in set $\{J_m, m + 1\}$. The conditions (a) and (b) clearly hold for $J_{m+1} = \{J_m, m + 1\}$, for it is impossible to construct such a set in $\{1, 2, \ldots, m + 1\}$ with the number of no-late jobs more than the number of no-late jobs in the set J_{m+1}. It is also clear that the last job has to be part of the set J_{m+1}, and the set has a minimum total processing time among sets with the same number of no-late jobs.

Case 2. Job $m+1$ is not completed by its due date in set $\{J_m, m+1\}$. From the fact that n_m is such a set of $\{1, 2, \ldots, m\}$ for which the maximum number of jobs to be completed on time and that set J_m has the smallest total processing time among sets with n_m on-time completions, we know that $n_{m+1} = n_m$. Adding job $m+1$ to set J_m does not increase the number of no-late jobs. And we know from Lemma 3 that adding job $m+1$ to the set and deleting the largest job among $\{J_m, m + 1\}$ keeps the same number of no-late jobs and reduces the total time it takes to process these jobs. It can be easily shown that J_{m+1} satisfies the conditions (a) and (b). The proof is completed. $\square$

Theorem 10 *For the problem $1|LE_t, p_j = p|\sum U_j$, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of d_j, i.e., the EDD rule is optimal.*

Theorem 11 *For the problem $1|LE_t, d_j = d|\sum U_j$, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of p_j, i.e., the SPT rule is optimal.*

If $d_j = kp_j$, the jobs have agreeable conditions, i.e., $p_i \leq p_j$ implies $d_i \leq d_j$ for all the jobs i and j. Hence, the following corollary can be easily obtained.

Corollary 2 *For the problem $1|LE_t, d_j = kp_j|\sum U_j$, an optimal schedule can be obtained by Moore's Algorithm.*

4 Conclusions

A type of scheduling problems with a time-dependent learning effect was studied in this paper. It was shown by several examples that the weighted sum of completion times minimization problem, the maximum lateness minimization problem and the number of tardy jobs minimization problem cannot be optimally solved by the corresponding classical scheduling rules, respectively. But for some special cases, the problems can be solved in polynomial time. We also used the classical rules as a heuristic algorithm for these three general problems, respectively, and analyzed their worst-case error bounds. Future research may focus on determining the complexity of these three problems as they remain open.

Acknowledgements

This research was supported in part by The Hong Kong Polytechnic University under grant number A628 from the *Area of Strategic Development in China Business Services*. The research of the first author was partially supposed by the foundation of Shenyang Institute of Aeronautical Engineering under grant number 05YB08.

References

[1] Bachman A, Janiak A. Scheduling jobs with position-dependent processing times. Journal of the Operational Research Society 2004; 55: 257-264.

[2] Badiru AB. Computational survey of univariate and multivariate learning curve models. IEEE transactions on Engineering Management 1992; 39: 176-188.

[3] Biskup D. Single-machine scheduling with learning considerations. European Journal of Operational Research 1999; 115: 173-178.

[4] Biskup D, Simons D. Common due date scheduling with autonomous and induced learning. European Journal of Operational Research 2004; 159: 606-616.

[5] Cheng TCE, Wang G. Single machine scheduling with learning effect considerations. Annals of Operations Research 2000; 98: 273-290.

[6] Jackson JR. Scheduling a production line to minimize maximum tardiness. Research Report 43, Management Sciences Research Project, UCLA, January, 1955.

[7] Kuo W-H, Yang D-L. Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect. European Journal of Operational Research (to appear).

[8] Kuo W-H, Yang D-L. Single-machine group scheduling with a time-dependent learning effect. Computers & Operations Research (to appear).

[9] Lee W-C, Wu C-C. Minimizing total completion time in a two-machine flowshop with a learning effect. International Journal of Production Economics 2004; 88: 85-93.

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

[11] Liu J, Sun S, He L. Some single machine scheduling problems with learning effect under consistent condition. OR Transactions 2003; 7 (3): 21-28.

[12] Moore J. An n job, one machine sequencing algorithm for minimizing the number of late jobs. Management Science 1968; 15: 102-109.

[13] Mosheiov G. Scheduling problems with a learning effect. European Journal of Operational Research 2001; 132: 687-693.

[14] Mosheiov G. Parallel machine scheduling with a learning effect. Journal of the Operational Research Society 2001; 52: 1165-1169.

[15] Mosheiov G, Sidney JB. Scheduling with general job-dependent learning curves. European Journal of Operational Research 2003; 147: 665-670.

[16] Mosheiov G, Sidney JB. Note on scheduling with general learning curves to minimize the number of tardy jobs. Journal of the Operational Research Society 2005; 56: 110-112.

[17] M. Pinedo. Scheduling Theory, Algorithms, and Systems, Prentice Hall, New Jersey, 1995.

[18] Wang J-B. Flow shop scheduling jobs with position-dependent processing times. Journal of Applied Mathematics and Computing 2005; 18: 383-391.

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

[20] Wright TP. Factors affecting the cost of airplanes. Journal of the Aeronautical Sciences 1936; 3: 122-128.