Single-machine scheduling problems with the time-dependent learning effect

Wen-Hung Kuo, Dar-Li Yang*

Department of Information Management, National Formosa University, Yun-Lin, 632, Taiwan

Received 27 March 2006; accepted 10 April 2006

Abstract

In this paper, three single-machine scheduling problems with time-dependent learning effect model to minimize the sum of weighted completion times, the sum of the $k$th powers of completion times and the maximum lateness are investigated. We show that the shortest processing time first (SPT) rule is an optimal solution for the problem to minimize the sum of the $k$th powers of completion times. Some polynomial time solutions are provided for the other two problems under certain conditions, respectively.

Keywords: Maximum lateness; Completion time; Learning effect; Scheduling; Polynomial time

1. Introduction

Recently, there has been growing interest in the literature to study scheduling problems with a learning effect. Biskup [1] was the first to investigate the learning effect in scheduling problems. He assumed a learning process that reflects a decrease in the processing time as a function of the number of repetitions of the production of a single item, i.e. as a function of the job position in the sequence. Biskup showed that the single-machine scheduling problem with the consideration of learning effects remain polynomially solvable for two objectives, namely minimizing the deviation from a common due date and minimizing the sum of flow-times. Mosheiov [2,3] followed Biskup’s learning model and studied some other scheduling problems. The assumption of learning effect in these studies is job-independent, i.e. the learning effect of a job only depends on its position in a sequence. However, in many realistic situations, the learning of some jobs may be better than that of others in a scheduling sequence. Therefore, Mosheiov and Sidney [4,5] further considered a more general learning effect model in which the learning effects of some jobs are better than those of others in a sequence, i.e. the learning effects are job-dependent. They showed that some scheduling problems with job-dependent learning effect remain polynomially solvable.

The studies mentioned above assumed that the learning is a function of job repetitions. However, in some realistic situations, the learning may depend on the total processing time of jobs previously executed. Therefore, Kuo and Yang [6,7] introduced a time-dependent learning effect model and incorporated it into single-machine scheduling problems with the objectives of minimizing the makespan and the total completion time.

* Corresponding author. Tel.: +886 5 631 5100; fax: +886 5 632 7291.
E-mail address: dlyang@nfu.edu.tw (D.-L. Yang).

© 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2006.04.035
In this paper, we study single machine scheduling problems with the learning effect model presented by Kuo and Yang [6,7] and consider some other performance measures. The rest of the paper is organized as follows. The problem description is presented in Section 2. In Section 3, some preliminary results are provided for the following analysis. The problems to minimize the sum of weighted completion times, the sum of the kth powers of completion times and maximum lateness are investigated in Sections 4, 5, and 6, respectively. Finally, some conclusions are given in the last section.

2. Problem description

There are n jobs to be processed on a single machine. Each of them is available at time zero. Let \( p_i \) denote the normal (sequence-independent) processing time of job \( i \) \((J_i, i = 1, 2, \ldots, n)\) in a sequence and \( p_{[k]} \) denote the normal processing time of a job if scheduled in the \( k \)th position in a sequence. As in Kuo and Yang [7], we assume that the actual processing time of job \( i \) if scheduled in position \( r \) is given as follows.

\[
p_{ir} = (1 + p_{[1]} + p_{[2]} + \cdots + p_{[r-1]})^a p_i = \left(1 + \sum_{k=1}^{r-1} p_{[k]}\right)^a p_i
\]

where \( a \leq 0 \) is a constant learning index and \( p_{[k]} \) is the normal processing time of a job if scheduled in position \( k \) in a sequence. In such a learning effect assumption, the actual processing time of a job is affected by the total normal processing time of the previous \((r - 1)\) jobs scheduled in front of it. The normal processing time of a job is incurred if the job is scheduled first in a sequence. The processing times of the following jobs are smaller than their normal processing times because of the time-dependent learning effect. For convenience, we denote the time-dependent learning effect given in Eq. (1) by \( LE_i \). In addition, for a given schedule \( q \), let \( C_i = C_i(q) \) and \( w_i \) denote the completion time and the corresponding weight of job \( i \), respectively. Let \( L_i = L_i(q) \) denote the lateness of job \( i \) where \( L_i = C_i - d_i \) and the maximum lateness \( L_{\max} = \max_i L_i = \max_i (C_i - d_i) \). The maximal completion time is denoted by \( C_{\max} = \max_i C_i \). Thus, using the three-field notation schema \( \alpha/\beta/\gamma \) introduced by Graham et al. [8], the problems to minimize the sum of weighted completion times, the sum of the \( k \)th powers of completion times and the maximum lateness in single-machine scheduling are denoted by \( 1/LE_i/\sum w_i C_i \), \( 1/LE_i/\sum C_i^k \) and \( 1/LE_i/L_{\max} \), respectively.

3. Preliminary results

First, one theorem and two useful lemmas are introduced in this section.

**Theorem 1.** For the problem \( 1/LE_i/C_{\max} \), there exists an optimal schedule in which jobs are sequenced in non-decreasing order of \( p_i \) (i.e. Shortest processing time rule, SPT rule).

**Proof.** See the proof in Kuo and Yang [7]. \( \square \)

The following lemmas were proposed by Kuo and Yang [9]. To make the lemmas easily for a reader to follow, we give the similar proofs of these lemmas as follows.

**Lemma 1.** \( 1 - \lambda_1 (1 + t)^a + \lambda_2 at (1 + t)^{a-1} \geq 0 \) if \( a \leq 0, t \geq 0 \) and \( 0 \leq \lambda_1 \leq \lambda_2 \leq 1 \).

**Proof.** Let \( h(t) = 1 - \lambda_1 (1 + t)^a + \lambda_2 at (1 + t)^{a-1} \). First, we calculate the first derivatives of \( h(t) \), and show that it is nonnegative for \( a \leq 0, t \geq 0 \) and \( 0 \leq \lambda_1 \leq \lambda_2 \leq 1 \). (hence \( h(t) \) is increasing for the above values of \( a, t, \lambda_1 \) and \( \lambda_2 \)). Then, for \( a \leq 0, t \geq 0 \) and \( 0 \leq \lambda_1 \leq \lambda_2 \leq 1 \) we have \( h'(t) = -\lambda_1 a (1 + t)^{a-1} + \lambda_2 a t (1 + t)^{a-1} + \lambda_2 a (a-1) t (1 + t)^{a-2} \geq 0 \). Next, since \( h(0) = 1 - \lambda_1 \geq 0 \), hence \( h(t) \) is increasing for \( a \leq 0, t \geq 0 \) and \( 0 \leq \lambda_1 \leq \lambda_2 \leq 1 \). Thus, we have \( 1 - \lambda_1 (1 + t)^a + \lambda_2 at (1 + t)^{a-1} \geq 0 \) for \( a \leq 0, t \geq 0 \) and \( 0 \leq \lambda_1 \leq \lambda_2 \leq 1 \). This completes the proof. \( \square \)

**Lemma 2.** \( \alpha (1 - \lambda_1 (1 + t)^a) - (1 - \lambda_2 (1 + \alpha t)^a) \geq 0 \) if \( \alpha \geq 1, t \geq 0, a \leq 0 \) and \( 0 \leq \lambda_1 \leq \lambda_2 \leq 1 \).
with respect to Let Theorem 1 Fig. 1, the result of part (a) follows. Thus, the proof is omitted here. We provide the proof of part (b) as follows.

Proof. Let

\[ f(\alpha) = \alpha(1 - \lambda_1(1 + t)^\alpha) - (1 - \lambda_2(1 + \alpha t)^\alpha). \]  

(2)

Taking the first and second derivatives of Eq. (2) with respect to \( \alpha \), we obtain

\[ f'(\alpha) = 1 - \lambda_1(1 + t)^\alpha + \lambda_2\alpha t(1 + \alpha t)^{\alpha-1} \]

(3)

and

\[ f''(\alpha) = \lambda_2\alpha(a - 1)t^2(1 + \alpha t)^{\alpha-2}. \]

(4)

Hence, \( f'(\alpha) \) is increasing on \( \alpha \geq 1, t \geq 0, a \leq 0 \) and \( 0 \leq \lambda_1 \leq \lambda_2 \leq 1 \) for \( f''(\alpha) \geq 0 \). In addition, from Lemma 1, we have

\[ f'(1) = 1 - \lambda_1(1 + t)^\alpha + \lambda_2\alpha t(1 + t)^{\alpha-1} \geq 0. \]

(5)

Therefore, \( f'(\alpha) \geq f'(1) \geq 0 \) for \( \alpha \geq 1, t \geq 0, a \leq 0 \) and \( 0 \leq \lambda_1 \leq \lambda_2 \leq 1 \). Hence, \( f(\alpha) \) is increasing on \( \alpha \geq 1, t \geq 0, a \leq 0 \) and \( 0 \leq \lambda_1 \leq \lambda_2 \leq 1 \). Also, \( f(\alpha) \geq f(1) = (\lambda_2 - \lambda_1)(1 + t)^\alpha \geq 0 \) for \( \alpha \geq 1, t \geq 0, a \leq 0 \) and \( 0 \leq \lambda_1 \leq \lambda_2 \leq 1 \). Therefore, we have \( \alpha(1 - \lambda_1(1 + t)^\alpha) - (1 - \lambda_2(1 + \alpha t)^\alpha) \geq 0 \) for \( \alpha \geq 1, t \geq 0, a \leq 0 \) and \( 0 \leq \lambda_1 \leq \lambda_2 \leq 1 \). This completes the proof. \( \square \)

4. The sum of weighted completion times minimization

In this section, we show that some polynomial time solutions are optimal for the problem \( 1/LE_1/ \sum w_i C_i \) under certain conditions.

Theorem 2. For the problem \( 1/LE_1/ \sum w_i C_i \), if jobs have agreeable weights, i.e. \( p_i \leq p_j \) implies \( w_i \geq w_j \) for all jobs \( J_i \) and \( J_j \), there exists an optimal schedule in which jobs are sequenced in non-decreasing order of \( p_i/w_i \) (i.e. Shortest weighted processing time rule, SWPT rule).

Proof. Let \( S_1 = (\pi_1, J_h, J_i, J_j, \pi_2) \) denote a sequence where \( J_i \) and \( J_j \) are scheduled in the \( r \)th and the \((r + 1)\)th positions. Moreover, let \( \pi_1 \) and \( \pi_2 \) denote the partial sequences of \( S_1 \) before and after \( J_h, J_i \) and \( J_j \), respectively. \( \pi_1 \) or \( \pi_2 \) may be empty. Let \( S_2 \) denote the same sequence with \( J_i \) and \( J_j \) mutually exchanged. The \( S_1 \) and \( S_2 \) sequences are shown in Fig. 1.

In order to prove that the sum of weighted completion times of the problem \( 1/LE_1/ \sum w_i C_i \) is minimized by the SWPT rule (i.e. \( p_i/w_i \leq p_j/w_j \)). It is sufficient to show that (a) \( C_j(S_1) \leq C_j(S_2) \), and (b) \( C_i(S_1) + C_j(S_1) \leq C_j(S_2) + C_i(S_2) \). Part (a) guarantees that all jobs scheduled in \( S_1 \) after the pair of \( J_i \) and \( J_j \), have completion times not larger than their completion times in \( S_2 \). Part (b) guarantees that the contribution to the sum of weighted completion times of \( J_i \) and \( J_j \) in sequence \( S_1 \) is less than or equal to their contribution in sequence \( S_2 \).

By Theorem 1, the result of part (a) follows. Thus, the proof is omitted here. We provide the proof of part (b) as follows.
First, we have
\[
 w_iC_i(S_1) + w_jC_j(S_1) = w_iC_h(S_1) + w_i p_i \left( 1 + \sum_{k=1}^{r-1} p[k] \right)^a + w_jC_h(S_1) + w_j p_j \left( 1 + \sum_{k=1}^{r-1} p[k] \right)^a \]
and
\[
 w_jC_j(S_2) + w_iC_i(S_2) = w_jC_h(S_2) + w_j p_j \left( 1 + \sum_{k=1}^{r-1} p[k] \right)^a + w_iC_h(S_2) + w_i p_i \left( 1 + p_i + \sum_{k=1}^{r-1} p[k] \right)^a \]

Then, since \( C_h(S_1) = C_h(S_2) \), we obtain
\[
 w_jC_j(S_2) + w_iC_i(S_2) - \left( w_iC_i(S_1) + w_jC_j(S_1) \right) = (w_i + w_j)(p_j - p_i) + w_i p_i (1 + p_j)^a - w_j p_j (1 + p_i)^a \quad \text{if } r = 1
\]
or
\[
 w_jC_j(S_2) + w_iC_i(S_2) - \left( w_iC_i(S_1) + w_jC_j(S_1) \right) = (w_i + w_j) p_j \left( 1 + \sum_{k=1}^{r-1} p[k] \right)^a + w_i p_i \left( 1 + p_i + \sum_{k=1}^{r-1} p[k] \right)^a - (w_i + w_j) p_i \left( 1 + \sum_{k=1}^{r-1} p[k] \right)^a + w_i p_i \left( 1 + p_i + \sum_{k=1}^{r-1} p[k] \right)^a \quad \text{if } r \geq 2.
\]

Case 1. \( r = 1 \)
Let \( \lambda_1 = \frac{w_j}{w_i + w_j}, \lambda_2 = \frac{w_i}{w_i + w_j} \) and \( \alpha = p_j / p_i \). Then dividing both sides of Eq. (8) by \( w_i + w_j \), we have
\[
 \frac{w_jC_j(S_2) + w_iC_i(S_2) - \left( w_iC_i(S_1) + w_jC_j(S_1) \right)}{w_i + w_j} = \left( p_j - p_i \right) + \lambda_2 p_i (1 + p_j)^a - \lambda_1 p_j (1 + p_i)^a = \alpha p_i (1 - \lambda_1 (1 + p_i)^a) - p_j (1 - \lambda_2 (1 + \alpha t)^a).
\]
If \( p_i \leq p_j \) and \( w_i \geq w_j \), then \( \alpha \geq 1 \) and \( 0 \leq \lambda_1 \leq \lambda_2 \leq 1 \). From Lemma 2, we have
\[
 \frac{w_jC_j(S_2) + w_iC_i(S_2) - \left( w_iC_i(S_1) + w_jC_j(S_1) \right)}{w_i + w_j} = p_i \left( \alpha (1 - \lambda_1 (1 + t)^a) - (1 - \lambda_2 (1 + \alpha t)^a) \right) \geq 0,
\]
where \( t = p_i > 0 \).
Consequently, since \( (w_i + w_j) > 0 \), \( w_iC_i(S_1) + w_jC_j(S_1) \leq w_jC_j(S_2) + w_iC_i(S_2) \) if \( p_i \leq p_j \) and \( w_i \geq w_j \).

Case 2. \( r \geq 2 \)
Let \( x = 1 + \sum_{k=1}^{r-1} p[k] \). Then dividing both sides of Eq. (9) by \( x^a (w_i + w_j) \), we obtain
\[
 \frac{w_jC_j(S_2) + w_iC_i(S_2) - \left( w_iC_i(S_1) + w_jC_j(S_1) \right)}{x^a (w_i + w_j)} = p_j + \left( \frac{w_i}{w_i + w_j} \right) p_i \left( 1 + \frac{p_j}{x} \right)^a - p_i - \left( \frac{w_j}{w_i + w_j} \right) p_j \left( 1 + \frac{p_i}{x} \right)^a.
\]
Again, let \( \lambda_1 = \frac{w_i}{w_i + w_j} \) and \( \lambda_2 = \frac{w_j}{w_i + w_j} \) and \( \alpha = \frac{p_j}{p_i} \). Then Eq. (10) can be rewritten as

\[
\frac{w_jC_j(S_2) + w_iC_i(S_2) - (w_jC_i(S_1) + w_jC_j(S_1))}{x^a(w_i + w_j)} = \alpha p_i - \lambda_1 \alpha p_i \left( 1 + \frac{p_i}{x} \right)^a - \left( p_i - \lambda_2 p_i \left( 1 + \alpha \left( \frac{p_i}{x} \right) \right)^a \right).
\]

If \( p_i \leq p_j \) and \( w_i \geq w_j \), then \( \alpha \geq 1 \) and \( 0 \leq \lambda_1 \leq \lambda_2 \leq 1 \). From Lemma 2, we have

\[
\frac{w_jC_j(S_2) + w_iC_i(S_2) - (w_jC_i(S_1) + w_jC_j(S_1))}{x^a(w_i + w_j)} = p_i \left( \alpha \left( 1 + \lambda_1 (1 + t)^a \right) - (1 - \lambda_2 (1 + \alpha t)^a) \right) \geq 0
\]

where \( t = \frac{p_i}{x} > 0 \).

Consequently, since \( x^a(w_i + w_j) > 0 \), \( w_jC_i(S_1) + w_jC_j(S_1) \leq w_jC_j(S_2) + w_iC_i(S_2) \) if \( p_i \leq p_j \) and \( w_i \geq w_j \).

This completes the proof of part (b) and thus of the theorem. \( \square \)

If the processing times of all jobs are equal, i.e. \( p_i = p \), then the following corollary can be easily obtained.

**Corollary 1.** For the problem \( 1/LE_i \), \( p_i = p/\sum w_iC_i \), there exists an optimal schedule in which jobs are sequenced in non-increasing order of \( w_i \).

5. The sum of the \( k \)th powers of completion times minimization

Townsend [10] studied a single machine scheduling problem with a quadratic cost function of completion times. His analysis implied that the problem \( 1/\sum C_i^k \) can be solved optimally by sequencing jobs in non-decreasing order of their basic processing times. In some scheduling situations, it is possible to consider a polynomial cost function of degree \( k \). Therefore, we consider a more general scheduling measure, that is, the sum of the \( k \)th powers of completion times. In addition, we show that the problem \( 1/LE_i/\sum C_i^k \) is still solved optimally by sequencing jobs in non-decreasing order of their basic processing times.

**Theorem 3.** For the problem \( 1/LE_i/\sum C_i^k \) where \( k \) is a positive real number, there exists an optimal schedule in which jobs are sequenced in non-decreasing order of \( p_i \) (i.e. Shortest processing time rule, SPT rule).

**Proof.** We use the same notations mentioned above. Hence, the completion times of \( J_i \) and \( J_j \) in sequence \( S_1 \) and \( S_2 \) are given as follows.

\[
C_i(S_1) = C_h(S_1) + p_i \left( 1 + \sum_{k=1}^{r-1} p[k] \right)^a,
\]

\[
C_j(S_1) = C_h(S_1) + p_i \left( 1 + \sum_{k=1}^{r-1} p[k] \right)^a + p_j \left( 1 + p_i + \sum_{k=1}^{r-1} p[k] \right)^a,
\]

\[
C_i(S_2) = C_h(S_2) + p_j \left( 1 + \sum_{k=1}^{r-1} p[k] \right)^a,
\]

and

\[
C_j(S_2) = C_h(S_2) + p_j \left( 1 + \sum_{k=1}^{r-1} p[k] \right)^a + p_i \left( 1 + p_j + \sum_{k=1}^{r-1} p[k] \right)^a.
\]

By comparing Eqs. (11) and (13), if \( p_i \leq p_j \), it is clear that \( C_i(S_1) \leq C_j(S_2) \) for \( C_h(S_1) = C_h(S_2) \). In addition, if \( p_i \leq p_j \), it follows from Theorem 1 that \( C_j(S_1) \leq C_i(S_2) \). Therefore, since \( k \) is a positive real number, we have \( C_i^k(S_1) \leq C_j^k(S_2) \) and \( C_j^k(S_1) \leq C_i^k(S_2) \). Consequently, \( \sum C_i^k(S_1) \leq \sum C_i^k(S_2) \). This completes the proof of the theorem. \( \square \)
6. The maximum lateness minimization

In this section, we also show that some polynomial time solutions are optimal for the problem $1/L_E_t/L_{\text{max}}$ under certain conditions.

**Theorem 4.** For the problem $1/L_E_t/L_{\text{max}}$, if jobs have agreeable due-dates, i.e. $p_i \leq p_j$ implies $d_i \leq d_j$ for jobs $J_i$ and $J_j$, there exists an optimal schedule in which jobs are sequenced in non-decreasing order of $d_i$ (i.e. Earliest due-date rule, EDD rule).

**Proof.** This theorem can be proved by a pairwise interchange of jobs. First, we still use the same notations mentioned above. Then we have

$$L_i(S_1) = C_i(S_1) + p_i \left(1 + \sum_{k=1}^{r-1} p_k\right)^a - d_i,$$

$$L_j(S_1) = C_j(S_1) + p_i \left(1 + \sum_{k=1}^{r-1} p_k\right)^a + p_j \left(1 + p_i + \sum_{k=1}^{r-1} p_k\right)^a - d_j,$$

$$L_j(S_2) = C_j(S_2) + p_j \left(1 + \sum_{k=1}^{r-1} p_k\right)^a - d_j$$

and

$$L_i(S_2) = C_i(S_2) + p_j \left(1 + \sum_{k=1}^{r-1} p_k\right)^a + p_i \left(1 + p_i + \sum_{k=1}^{r-1} p_k\right)^a - d_i.$$

If $d_i \leq d_j$, then we obtain $L_j(S_2) \leq L_i(S_1)$. That is, $L_i(S_2) = \max\{L_j(S_2), L_i(S_1)\}$ if $d_i \leq d_j$.

Hence, if $p_i \leq p_j$, from Theorem 1, the completion time of $C_j(S_1)$ is less than or equal to $C_i(S_2)$. Since $C_h(S_1) = C_h(S_2)$, if $p_i \leq p_j$ and $d_i \leq d_j$, we have $L_j(S_1) \leq L_i(S_2)$ and $L_i(S_1) \leq L_i(S_2)$. Therefore, if $p_i \leq p_j$ and $d_i \leq d_j$, then we have $\max\{L_i(S_1), L_j(S_1)\} \leq \max\{L_j(S_2), L_i(S_2)\}$. This completes the proof of the theorem. $\square$

If the processing times of all jobs are equal, i.e. $p_i = p$, then the following corollary can be easily obtained.

**Corollary 2.** For the problem $1/L_E_t$, $p_i = p/L_{\text{max}}$, there exists an optimal schedule in which jobs are sequenced in non-decreasing order of $d_i$ (i.e. Earliest due-date rule, EDD rule).

If $d_i = kp_i$, then all jobs have agreeable due-dates, i.e. $p_i \leq p_j$ implies $d_i \leq d_j$ for jobs $J_i$ and $J_j$. Then the following corollary can be easily obtained.

**Corollary 3.** For the $1/L_E_t$, $d_i = kp_i/L_{\text{max}}$ problem, there exists an optimal schedule in which jobs are sequenced in non-decreasing order of $d_i$ (i.e. Earliest due-date rule, EDD rule).

If all jobs share a common due-date, i.e. $d_i = d$, then the following corollary can be easily obtained.

**Corollary 4.** For the $1/L_E_t$, $d_i = d/L_{\text{max}}$ problem, there exists an optimal schedule in which jobs are sequenced in non-decreasing order of $p_i$ (i.e. Shortest processing time rule, SPT rule).

7. Conclusions

In this paper, we study single machine scheduling problems with the time-dependent learning effect model. Three scheduling problems $1/L_E_t/\sum w_iC_i$, $1/L_E_t/\sum C_i^k$, and $1/L_E_t/L_{\text{max}}$ are considered. We show that SPT rule is an optimal solution for the problem $1/L_E_t/\sum C_i^k$. Moreover, some polynomial time solutions are provided for the problems $1/L_E_t/\sum w_iC_i$ and $1/L_E_t/L_{\text{max}}$ under some certain conditions, respectively. We note that the complexity of $1/L_E_t/\sum w_iC_i$ and $1/L_E_t/L_{\text{max}}$ remains open. It will be an interesting topic for future research.
Acknowledgement

This research was supported in part by the National Science Council of Taiwan, Republic of China, under grant number NSC-94-2213-E-150-016.

References