SINGLE MACHINE AND FLOWSHOP SCHEDULING PROBLEMS WITH SUM-OF-PROCESSING TIME BASED LEARNING PHENOMENON

XINGONG ZHANG
1. College of Mathematics Science
Chongqing Normal University, Chongqing, 401331, China
2. Key Laboratory for Optimization and Control
Ministry of Education, Chongqing, China

(Communicated by Ada Che)

ABSTRACT. This paper addresses single machine and flowshop machines with the learning phenomenon. The learning phenomenon means that the actual jobs processing time of a job is a non-increasing function of the sum of processing times of jobs already processed. Under single machine, some properties firstly are presented to solve the objectives of minimizing the makespan problem, the total (weighted) completion time problem, the maximum lateness problem and the total tardiness problem. We show that minimizing the makespan problem and the total completion time problem can be solved in polynomial time. For the weighted completion time problem, the maximum lateness problem and the total tardiness problem, we give heuristic algorithm based on the corresponding optimal schedule and analyze the worst case error bound. Furthermore, we also show that the three problems are polynomially solvable under certain conditions. Under flowshop machines, we finally show that the makespan problem and the total completion time problem under more specialized proportional job processing times can be solved in polynomial time.

1. Introduction. In the classical scheduling problems, it is assumed that job processing times are fixed and known. However, the assumption might be unrealistic in many situations. Based on repeating similar or identical tasks, the production efficiency of the workers or processors will be improved because of the obtained experience. The phenomenon is generally called learning effect in literatures. The learning environment is first discussed in scheduling fields by Biskup [1]. Biskup [1] presented the learning phenomenon in which the actual processing time of the job is decreasing function of job scheduled positions, and showed that minimizing the deviation from a common due date and minimizing the sum of flow times the scheduling can remain polynomially solvable. Biskup[2] reviewed comprehensively related learning effect models on various machine phenomenons. Two different study approaches to learning effect in machine scheduling environments are mentioned as
follows: position-based learning effect and the sum-of-processing-time based learning effect. Wang\cite{24} considered the single-machine scheduling problems with the effects of learning effect and deterioration jobs. He showed that the makespan and the sum of completion times (square) minimizing problems remain polynomially solvable, respectively. Kuo and Yang\cite{13} presented sum-of-processing-time based learning effect model where the actual processing time of a job is a function of the total processing time of jobs scheduled in front of it. Yin et al.\cite{31} developed a more general sum-of-processing-time based learning effect model. They showed that some single-machine scheduling problems and $m$-machine permutation flow-shop problems can be solved in a polynomial time. Ji et al.\cite{10} combined DeJongs learning effect with deteriorating jobs on parallel machines. They showed that minimizing makespan problem is NP-hard, and provided a $(1+\varepsilon)$ fully polynomial-time approximation scheme in $O(n^{3m+1}L^{2m+1}\varepsilon^{2m})$ time, where $m$ denotes the number of the machines and $L$ is a given constant. He et al.\cite{5} studied that the processing time of a job is a function of the amount of allocated resource, the general job-dependent learning effect and the job-dependent control parameter. The bounded linear and convex resource consumption functions. The corresponding polynomial time algorithms are also developed to find the optimal schedule of several versions of these problems. For most research results on other scheduling models with learning effects and different machine environments, one can refer to \cite{8}, \cite{9}, \cite{26}, \cite{29}, \cite{30}, \cite{34}.

In addition, Wang and Cheng\cite{25} considered scheduling problems with a volume-dependent piecewise linear processing time function learning effects on a single machine. They obtained that the maximum lateness problem is NP-hard in the strong sense by transforming from the 3-partition problem. Koulamas and Kyparisis\cite{12} considered a general learning effect function on a single machine, where the learning effect is a function of the sum-of-processing-time of the already processed jobs, i.e., if job $j$ is scheduled in position $r$ in job sequence, its actual processing time is

\[ p_{jr} = p_j(1 - \frac{\sum_{i=1}^{r-1} P_i}{\sum_{i=1}^{r} p_i})^a, \]

where $a \leq 1$ is the learning index. They showed that the makespan problem and total completion time problem can be obtained the optimal job sequence by SPT rule. Huang et al. \cite{6} studied the following learning effect model: $p_{jr} = p_j(1 + \sum_{i=1}^{r-1} p_{i})^a b^{r-1}$, where $a (> 1)$ and $b (0 < b \leq 1)$ denote deteriorated rate and learning rate. In another paper of Huang et al.\cite{7}, they presented the sum-of-processing-time based learning effect model with setup time considerations on a single machine. The setup time is proportional to the total processing times of the already processed jobs, i.e., the setup time is called by past-sequence-dependent setup time. The objective functions conclude as follows: the makespan, the total (weighted) completion time, the sum of the $th (\geq 0)$ power of the completion times and the maximum lateness. Yin et al.\cite{32} introduced scheduling models with position-dependent learning effect and time-dependent deterioration simultaneously on a single machine. The objective functions are to minimize the makespan, the total (weighted) completion time, the discounted total weighted completion time, and the maximum lateness, respectively. Cheng et al.\cite{4} introduced a position-weighted learning effect model based on sum-of-logarithm-processing times and jobs position
on a single machine. They provided optimal solutions to minimize the makespan and the total completion time. They also presented some polynomial time algorithms under an agreeable situation to minimize the weighted completion times, the maximum lateness and the total tardiness, respectively. Lu and Wang [22] combined jobs position and sum-of-processing-time based processing times on a single machine. The actual processing time of a job is function of its position and the total normal processing time of jobs already processed. For the makespan problem and the total completion time problem, they provided some optimal solutions in polynomial time under some special conditions. For most research results on other scheduling models with sum-of-processing time based learning effects, one can refer to [21] and [27].

In many realistic situations, both the machine and human learning effects might exist simultaneously. However, they might exist at the same time. For example, artificial intelligence robots are used in computers, motor vehicles, and many assembly lines. The actions of a robot are constantly modified through self-learning in processing the jobs. On the other hand, the operators in the control center learn how to give the commands efficiently through working experience. Based on above motivation, this paper considers position-weighted learning effect with combining sum-of-processing times and jobs position on a single machine. We consider two machine environments: single machine and flowshop machines. Our objectives are to minimize the makespan problem, the total completion time problem, the total weighted completion time problem, the maximum lateness problem and the total tardiness problem. The remaining part of this paper is organized as follows. In Section 2, we give the proposal Problem description. In Section 3, we analyze some optimization and several heuristic rules to give their worst case error bounds on a single machine. In Section 4, two-machine flowshop problems which the jobs have ordered or proportional processing times are considered. We show that the SPT sequence is optimal under the proposed model. The last section concludes this paper.

2. Problem description. Recently, Wu and Lee [28] and Lee and Wu [18] proposed learning models which consider both the machine and human learning effects, simultaneously. The model in [28] is described as follows. The actual processing time of job $j$ when scheduled in the $r$th position is as follows:

$$p_{j[r]} = p_j q_r^{a_1} (c_0 + \sum_{l=1}^{r-1} \beta_{r-l} p[l])^{a_2}$$

where $p_j$ is the normal processing time of job $j$, $p[l]$ is the normal processing time of a job scheduled in the $l$th position, $c_0 > 0$ is a given constant, $q_r$ is a non-decreasing function of the $r$th position, $\beta_i (i = 1, 2, \cdots, n)$ is a non-decreasing sequence of coefficients, $a_1 \leq 0$ and $a_2 \leq 0$ are the learning indices.

Lee and Wu [18] addressed the similar model of [28], the actual processing time of job $j$ when scheduled in the $r$th position is as follows:

$$p_{j[r]} = p_j (q(r) + \sum_{l=1}^{r-1} \beta_{r-l} p[l])^\alpha$$

where $q_r$ is a non-decreasing function of the $r$th job position and $\alpha < 0$ are the learning indices.
Lee et al. [19], [20] and Kuo and Yang [15] pointed out that the main results in recent papers [18] and [28] are incorrect, respectively. Especially, Kuo and Yang [14] given some counter examples. They deemed that there are some points need to be clarified in the proposed learning effect model. It is very much regret that they did not give the proof of sufficient condition.

Motivated by the idea of Kuo and Yang [15], we improve their presented models, so that the results still are valid in our new models. We develop an improved model as following: The learning phenomenon means that the actual processing time of job \( j \) scheduled in \( r \)th position is given by

\[
p_{j[r]} = p_{j}q_{r}^{a_1}(1 + \sum_{i=1}^{r-1} \beta_{i}p_{[i]})^{a_2},
\]

where \( p_{[i]} \) is the normal processing time of the job scheduled in the \( l \)th position, \( q_{r} \) is a non-decreasing function of the position \( r \), \( \beta_{i}(i = 1, 2, \cdots, n) \) is a non-decreasing sequence of coefficients (Note that two symbols only are mathematical abstractions), \( a_1 < 0 \) and \( a_2 < 0 \) are the learning indices.

For a given schedule \( S = \{1, 2, \cdots, n\} \), let \( C_{j} = C_{j}(S) \) represent the completion time of job \( j \). Also \( C_{max} = max\{C_{j}|j = 1, \cdots, n\}\), \( \sum C_{j} \), \( \sum w_{j}C_{j} \), \( L_{max} = max\{C_{j} - d_{j}|j = 1, \cdots, n\} \) and \( \sum T_{j} \) where \( T_{j} = max\{C_{j} - d_{j}, 0\} \) represent the makespan, the total (weight) completion, the maximum lateness and the total tardiness, respectively.

3. Single-machine problems under our improved learning effect. Next, we give two lemmas to solve the proposed learning effect problems. The proofs of the lemmas can be obtained by differentiation.

**Lemma 3.1.** \(-1 + q^{a_1}(1 + x)^{a_2} - xa_2q^{a_1}(1 + x)^{a_2-1} \leq 0, \text{ for } q \geq 1, \ x \geq 0 \text{ and } a_2 < 0.\)

**Lemma 3.2.** \(1 - \lambda + \lambda q^{a_1}(1 + x)^{a_2} - q^{a_1}(1 + \lambda x)^{a_2} \leq 0, \\text{ for } \lambda \geq 1, \ q \geq 1, \ x \geq 0 \text{ and } a_2 < 0.\)

**Theorem 3.1.** For the \( 1 | p_{j[r]} = p_{j}q_{r}^{a_1}(1 + \sum_{i=1}^{r-1} \beta_{i}p_{[i]})^{a_2} | C_{max} \) problem, an optimal schedule is obtained by the smallest processing time first rule (SPT rule).

**Proof.** Suppose that \( p_{i} \leq p_{j} \). Let \( S = (\pi \ i \ j \ \pi') \) and \( S' = (\pi \ i \ \pi') \) be two job schedules, where \( \pi \) and \( \pi' \) denote the partial job sequences in which they may be empty set. Furthermore, assume that there exist \( r - 1 \) jobs in partial sequence \( \pi \) and the completion time of the last job in partial sequence \( \pi \) is \( A \). To show that \( S \) dominates \( S' \), it suffices to show that the two following problems are true: (a) \( C_{j}(S) \leq C_{i}(S') \) and (b) \( C_{m}(S) \leq C_{m}(S') \) for any job \( m \) in \( \pi' \).

Under sequence \( S \), the completion times of job \( i \) and job \( j \) are, respectively,

\[
C_{i}(S) = A + p_{i}q_{r}^{a_1}(1 + \sum_{l=1}^{r-1} \beta_{l}p_{[l]})^{a_2},
\]

and

\[
C_{j}(S) = A + p_{j}q_{r}^{a_1}(1 + \sum_{l=1}^{r-1} \beta_{l}p_{[l]})^{a_2} + p_{j}q_{r+1}^{a_1}(1 + \sum_{l=1}^{r-1} \beta_{l}p_{[l]} + \beta_{r}p_{r})^{a_2}. \]

Similarly, the completion times of job \( i \) and job \( j \) in sequence \( S' \) are, respectively,

\[
C_{j}(S') = A + p_{j}q_{r}^{a_1}(1 + \sum_{l=1}^{r-1} \beta_{l}p_{[l]})^{a_2},
\]

and

\[
C_{i}(S') = A + p_{j}q_{r}^{a_1}(1 + \sum_{l=1}^{r-1} \beta_{l}p_{[l]})^{a_2} + p_{j}q_{r+1}^{a_1}(1 + \sum_{l=1}^{r-1} \beta_{l}p_{[l]} + \beta_{r}p_{r})^{a_2}. \]

The difference between Equations (3) and (5) is calculated as follows.
\[
C_j(S) - C_j(S') = (p_i - p_j)q_r^{a_1}(1 + \sum_{i=1}^{r-1} \beta_i p_i)[a_2 + p_j q_r^{a_1}(1 + \sum_{i=1}^{r-1} \beta_i p_i + \beta_r p_i)]^2
\]
\[
- p_i q_r^{a_1}(1 + \sum_{i=1}^{r-1} \beta_i p_i + \beta_r p_j)[a_2(1 + \sum_{i=1}^{r-1} \beta_i p_i)]^2
\]
\[
= p_i q_r^{a_1}(1 + \sum_{i=1}^{r-1} \beta_i p_i)[a_2 - \frac{p_j}{p_i} + \frac{p_j}{p_i} q_r^{a_1}(1 + \sum_{i=1}^{r-1} \beta_i p_i)]^2 \frac{\beta_r p_j}{1 + \sum_{i=1}^{r-1} \beta_i p_i}[a_2^2]
\]
By substituting \( \lambda = \frac{p_j}{p_i} \), \( q = \frac{q_r + 1}{p_i} \) and \( x = \frac{\beta_r p_j}{1 + \sum_{i=1}^{r-1} \beta_i p_i} \) into Equation (6), it is simplified to
\[
p_i q_r^{a_1}(1 + \sum_{i=1}^{r-1} \beta_i p_i)^2[1 - \lambda + \lambda \beta x(1 + x) a_2 - q a_1(1 + \lambda x) a_2].
\]
Note that \( p_i \leq p_j \), then we have \( \lambda \geq 1, q \geq 1, x \geq 0 \). From Lemma 3.2, we have \( C_j(S) \leq C_j(S') \). Therefore, part (a) holds. Next, the proof of part (b) will be given as follows.
\[
C_k(S) = C_j(S) + p_k q_r^{a_1}(1 + \sum_{i=1}^{r-1} \beta_i p_i + \beta_r p_i + \beta_{r+1} p_i)^2,
\]
and
\[
C_k(S') = C_j(S') + p_k q_r^{a_1}(1 + \sum_{i=1}^{r-1} \beta_i p_i + \beta_r p_i + \beta_{r+1} p_i)^2,
\]
where \( k \in \pi' \). Because \( (\beta_r p_i + \beta_{r+1} p_i) - (\beta_r p_j + \beta_{r+1} p_i) = (\beta_r - \beta_{r+1})(p_i - 1) \geq 0 \) for \( p_i \leq p_j \) and \( \beta_r \leq \beta_{r+1} \), we have \( C_k(S) \leq C_k(S') \). In other words, the completion time of the first job under sequence \( p_i \) is earlier than that in \( \pi' \). Similarly, we can show that the completion time of any job under sequence \( S \) is not larger than that of under sequence \( S' \). Therefore, \( C_{\text{max}}(S) \leq C_{\text{max}}(S') \).

**Theorem 3.2.** For the \( 1 \mid p_{j[r]} = p_j q_r^{a_1}(1 + \sum_{i=1}^{r-1} \beta_i p_i)\mid \sum C_j \) problem, an optimal schedule is obtained by sequencing jobs in non-decreasing order of \( p_j \) (SPT rule).

**Proof.** We use the same notation in the proof of Theorem 2.1. It suffices to show that \( \sum_{i=1}^{n} C_i(S) \leq \sum_{i=1}^{n} C_i(S') \). Taking the difference between Eq. (2) and (4), we have
\[
C_i(S) - C_i(S') = (p_i - p_j)q_r^{a_1}(1 + \sum_{i=1}^{r-1} \beta_i p_i)^2 \leq 0.
\]
By Theorem 3.1, we have \( C_j(S) \leq C_j(S') \) and \( C_m(S) \leq C_m(S') \) for any job \( m \) in \( \pi' \). Then we have \( \sum_{i=1}^{n} C_i(S) \leq \sum_{i=1}^{n} C_i(S') \). This completes the proof.

The total weighted completion time minimization problem denoted by \( 1\mid \sum w_j C_j \) can be solved by the weighted smallest processing time first (WSPT) rule [23], i.e., sequencing jobs in non-decreasing order of \( p_j/w_j \). The following example shows that the WSPT rule does not necessarily lead to an optimal schedule for the \( 1\mid p_{j[r]} = p_j q_r^{a_1}(1 + \sum_{i=1}^{r-1} \beta_i p_i)^2 \mid \sum w_j C_j \) problem.

**Example.** Let \( q_r = r \). Given \( n = 2, p_1 = 4, p_2 = 3, \alpha_1 = \alpha_2 = -0.2, w_1 = \frac{3}{2}, w_2 = 1, \beta_1 = 0.25 \) and \( \beta_2 = 0.5 \). The job sequence \( (1, 2) \) based on the WSPT rule
yields a value of 12.273574848, while the job sequence (2, 1) yields the optimal value of 12.170223252.

For the 1 | pj|r = pjqσi(1 + \sum_{l=1}^{r-1} \beta pj[l])qσi | \sum wjCj problem, we will use the WSPT rule as a heuristic algorithm. The performance of the algorithm is evaluated by the worst case error bound.

**Theorem 3.3.** Assume that S* is an optimal schedule and S = \{1, 2, \ldots, n\} is a feasible schedule for the problem 1 | pj|r = pjqσi(1 + \sum_{l=1}^{r-1} \beta pj[l])qσi | \sum wjCj according to the WSPT rule. Then its worst case error bound \( \rho_1 \) is

\[
\rho_1 = \frac{1}{q_n^{a_1}(1 + \beta_n \sum_{l=1}^{n} p_l)^{a_2}}.
\]

**Proof.** Since \( q_r \) is a non-decreasing function of the position \( r \), then

\[
\sum wjCj(S) = \sum_{r=1}^{n} wj_{[r]}Cj_{[r]}(S) = \sum_{r=1}^{n} wj_{(r)}(\sum_{k=1}^{r} p[j]q^{a_1}(1 + \sum_{l=1}^{k-1} \beta p[j[l]]q^{a_2})
\leq \sum_{r=1}^{n} wj_{(r)}(\sum_{k=1}^{r} p[j])
\]

where \( \sum_{r=1}^{n} wj_{(r)}(\sum_{k=1}^{r} p[j]) \) is the optimal value of the problem 1\|\sum wjCj.

Let \( S^* = \{S*(1), S*(2), \ldots, S*(n)\} \) be the optimal job sequence, where \( S*(1), S*(2), \ldots, S*(n) \) is a permutation of 1, 2, \ldots, n. Then

\[
\sum wjCj(S^*) = \sum_{r=1}^{n} wj_{(r)}Cj_{(r)}(S^*)
= \sum_{r=1}^{n} wj_{(r)}(\sum_{k=1}^{r} p[j]q^{a_1}(1 + \sum_{l=1}^{k-1} \beta p[j[l]]q^{a_2})
\geq \sum_{r=1}^{n} wj_{(r)}(\sum_{k=1}^{r} p[j]q^{a_1}(1 + \beta_n \sum_{l=1}^{n} p[l])^{a_2}.
\]

Hence, \( \rho_1 = \frac{1}{q_n^{a_1}(1 + \beta_n \sum_{l=1}^{n} p[l])^{a_2}} \). It is not difficult to see that \( \rho_1 = 1 \) if the learning effect does not exit, i.e., \( a_1 = a_2 = 0 \). This result is obvious because of the optimality of WSPT rule.

Although it is regretful that the WSPT rule does not provide the optimal job sequence for the 1 | \sum wjCj problem, the WSPT rule is still optimal job sequence if the jobs satisfy reversely agreeable weights, i.e., \( p_i \leq p_j \) implies \( w_i \geq w_j \) for all job \( i \) and job \( j \).

**Theorem 3.4.** For the 1 | pj|r = pjqσi(1 + \sum_{l=1}^{r-1} \beta pj[l])qσi | \sum wjCj problem, an optimal job sequence can be obtained by sequencing jobs in non-decreasing order of \( \frac{pj}{wj} \), i.e., WSPT rule, if the jobs satisfy reversely agreeable weights.

**Proof.** Similar to the proof of Theorem 3.1, assume that \( \frac{pj}{p_l} \geq \frac{w_j}{w_l} \). Based on the fact of reversely agreeable weights, i.e., \( p_i \leq p_j \) implies \( w_i \geq w_j \). Hence, we can obtain \( C_i[S] \leq C_i[S'] \) for \( 1 \leq l \leq n \), \( C_i(S) \leq C_j(S') \) and \( C_j(S) \leq C_i(S') \). To show
that $S$ dominates $S'$, we will show that $w_1C_i(S) + w_2C_j(S) \leq w_2C_j(S') + w_1C_i(S')$. Since
\[
w_1C_i(S) + w_2C_j(S) - (w_2C_j(S') + w_1C_i(S')) \\
\leq w_1C_i(S) + w_2C_j(S) - (w_2C_j(S) + w_1C_i(S)) \\
= (w_1 - w_2)(C_i(S) - C_j(S)).
\]

Note that $w_i \geq w_j$, $C_i(S) \leq C_j(S)$, then $w_1C_i(S) + w_2C_j(S) \leq w_2C_j(S') + w_1C_i(S')$. We show that $S$ dominates $S'$. Thus, repeating this interchange argument for all jobs not sequenced by the WSPT rule yields the proposed conclusion. □

The maximum lateness minimization problem on a single machine denoted by $1 \mid \mid \text{L}_{\text{max}}$ can be solved by the smallest due date first (EDD) rule[23], i.e., sequencing jobs in non-decreasing order of $d_j$. The following example will illustrate that the EDD rule does not necessarily generate an optimal job sequence for the $1 \mid \mid \text{L}_{\text{max}}$ problem.

**Example.** Let $q_r = r$. Given $n = 2$, $p_1 = 6$, $p_2 = 4$, $\alpha_1 = -0.2$, $\alpha_2 = -0.3$, $d_1 = 1.8$, $d_2 = 2$, $\beta_1 = 0.25$ and $\beta_1 = 0.5$. The sequence $(1, 2)$ from the EDD rule yields a value of 6.645282, while the sequence $(2, 1)$ yields the optimal value of 6.44264.

Next, we will find a worst case performance ratio to solve the presented problem approximately by using the EDD rule as a heuristic algorithm. It is obvious that the problems caused by non-positive $L_{\text{max}}$ will have to be avoided. Based on the idea of Kise et al.[11] and Cheng et al.[3], the maximum lateness $L_{\text{max}}$ will be added a constant value at least as large as the maximum due date $d_{\text{max}}$. Furthermore, the worst case error bound can be determined by the following ratio
\[
\rho_2 = \frac{L_{\text{max}}(S) + d_{\text{max}}}{L_{\text{max}}(S^*) + d_{\text{max}}},
\]
where and $L_{\text{max}}(S)$ and $L_{\text{max}}(S^*)$ denote the maximum lateness under job sequence $S$ and the optimal schedule $S^*$, and $d_{\text{max}} = \max\{d_j \mid j = 1, 2, \ldots, n\}$.

**Theorem 3.5.** For the problem $1\mid p_{j}[r] = p_jq_k^{a_1}(1 + \sum_{i=1}^{r-1} \beta_ip_{[i]}^{a_2}) \mid \text{L}_{\text{max}}$, let $S^*$ is an optimal job sequence and $S = \{(1), (2), \ldots, (n)\}$ is a job schedule by the EDD rule. Then the worst case error bound $\rho_2$ of the proposed problem is
\[
\rho_2 = 2 - \frac{1}{(1 + \beta n \sum_{i=1}^{n} p_i)^{a_2}}.
\]

**Proof.** Since $q_r$ is a non-decreasing function of position $r$, then
\[
L_{\text{max}}(S) = \max\{\sum_{k=1}^{r} p_{[k]}q_k^{a_1}(1 + \sum_{l=1}^{r-1} \beta_ip_{[l]}^{a_2} - d_{[r]} \mid r = 1, 2, \ldots, n\}
\leq \max\{\sum_{k=1}^{r} p_{[k]} - d_{[r]} \mid r = 1, 2, \ldots, n\}
\]
where $\max\{\sum_{k=1}^{r} p_{[k]} - d_{[r]} \mid r = 1, 2, \ldots, n\}$ is the optimal value of the class scheduling problem $1\mid \mid \text{L}_{\text{max}}$. 

Let \( \{S^*(1), S^*(2), \ldots, S^*(n)\} \) be the job sequence in the optimal schedule \( S^* \), where \( S^*(1), S^*(2), \ldots, S^*(n) \) is a permutation of 1, 2, \ldots, n. Then

\[
L_{\text{max}}(S^*) = \max \left\{ \sum_{k=1}^{r} p_{S^*(k)} q_k^{a_1} (1 + \sum_{l=1}^{r-1} \beta_l p_{S^*(l)})^{a_2} - d_{S^*(r)} \mid r = 1, 2, \ldots, n \right\}
\]

\[
= \max \left\{ \sum_{k=1}^{r} p_{S^*(k)} q_k^{a_1} (1 + \sum_{l=1}^{r-1} \beta_l p_{S^*(l)})^{a_2} + \sum_{k=1}^{r} p_{S^*(k)} - d_{S^*(r)} - \sum_{k=1}^{r} p_{S^*(k)} \mid r = 1, 2, \ldots, n \right\}
\]

\[
\geq \max \left\{ \sum_{k=1}^{r} p_{S^*(k)} - d_{\text{max}} + \sum_{k=1}^{r} p_{S^*(k)} q_k^{a_1} (1 + \beta_n \sum_{l=1}^{n} p_l)^{a_2} - 1 \right\}
\]

Hence, \( L_{\text{max}}(S) - L_{\text{max}}(S^*) \leq \sum_{k=1}^{n} p_{S^*(k)} q_k^{a_1} (1 + \beta_n \sum_{l=1}^{n} p_l)^{a_2} - 1 \) and

\[\rho_2 = \frac{L_{\text{max}}(S) + d_{\text{max}}}{L_{\text{max}}(S^*) + d_{\text{max}}} \leq 1 + \frac{L_{\text{max}}(S) - L_{\text{max}}(S^*)}{L_{\text{max}}(S^*) + d_{\text{max}}} \leq 2 - \frac{1}{(1 + \beta_n \sum_{l=1}^{n} p_l)^{a_2}}.\]

It is not difficult to see that \( \rho_2 = 1 \) if the learning effect does not exist, i.e., \( a_1 = a_2 = 0 \). It is intuitive because the optimal job sequence can be obtained by the EDD rule.

Although an optimal schedule for the problem 1 \( |p_j| = p_j q_j^{a_1} (1 + \sum_{l=1}^{r-1} \beta_l p_l)^{a_2} \mid L_{\text{max}} \) can not be obtained by the EDD rule. Applying the similar technique of the proof in Theorem 3.1 and 3.3, the following results can be presented.

**Theorem 3.6.** For the 1 \( |p_j| = p_j q_j^{a_1} (1 + \sum_{l=1}^{r-1} \beta_l p_l)^{a_2} \mid L_{\text{max}} \) problem, an optimal schedule is obtained by sequencing jobs in non-decreasing order of \( d_j \), if the processing time and the due date of the jobs satisfies agreeable order, i.e., \( p_i \leq p_j \Rightarrow d_i \leq d_j \).

Note that the class scheduling problem 1 || \( \sum T_j \) can be solved by the EDD rule, i.e., sequencing jobs in non-decreasing order of \( d_j \). The following example can illustrate that the problem 1 \( |p_j| = p_j q_j^{a_1} (1 + \sum_{l=1}^{r-1} \beta_l p_l)^{a_2} \mid \sum T_j \) can not be obtained the optimal job sequence by the EDD rule.

**Example.** Let \( q_r = r \). Given \( n = 2, p_{11} = 6, p_2 = 4, a_1 = -0.2, a_2 = -0.3, d_1 = 1.8, d_2 = 2, \beta_1 = 0.25 \) and \( \beta_1 = 0.5 \). The sequence \( (1, 2) \) from the EDD rule can generate the value of 10.65282, while the sequence \( (2, 1) \) can yield the optimal value of 8.44264.

Next, we will solve the proposed problem approximately by using the EDD rule as an heuristic algorithm. To find a worst case performance ratio, the problems caused by non-positive \( \sum T_j \) have to be avoided. Similar to Theorem 3.5, the worst case error bound can be determined by the ratio \( \rho_3 = \frac{\sum T_j(S) + nd_{\text{max}}}{\sum T_j(S^*) + nd_{\text{max}}} \), where
Theorem 3.7. Assume that \( S^* \) is an optimal job sequence and \( S = \{1, 2, \ldots, n\} \) is a job sequence according to the EDD rule under the problem \( 1|\sum_j p_j| \sum_j T_j \). Then

\[
\rho_3 = \frac{1}{q_n a_1 (1 + \beta_n \sum_{j=1}^n p_j) a_2}.
\]

Proof. Since \( q_r \) is a non-decreasing function of position \( r \), then

\[
\sum_{r=1}^n \{ \sum_{k=1}^r p_k | q_k a_1 (1 + \sum_{l=1}^{r-1} \beta_l p_l) a_2 - d_{(r)} \} \leq \sum_{r=1}^n \{ \sum_{k=1}^r p_k - d_{(r)} \}
\]

where \( \sum_{r=1}^n \{ \sum_{k=1}^r p_k - d_{(r)} \} \) is the optimal value of the class scheduling problem \( 1||\sum_j T_j \).

Let \( \{S^*(1), S^*(2), \ldots, S^*(n)\} \) be the job sequence under the optimal sequence \( S^* \), where \( S^*(1), S^*(2), \ldots, S^*(n) \) is a permutation of 1, 2, \ldots, \( n \). Then

\[
\sum_{j=1}^n \sum_{k=1}^r p_{S^*(k)} q_k a_1 (1 + \sum_{l=1}^{r-1} \beta_l p_{S^*(l)}) a_2 - d_{S^*(r)} \geq \sum_{r=1}^n \{ \sum_{k=1}^r p_{S^*(k)} - d_{S^*(r)} \}
\]

Hence, \( \sum_{j=1}^n \sum_{k=1}^r p_{S^*(k)} q_k a_1 (1 + \sum_{l=1}^{r-1} \beta_l p_l) a_2 - \sum_{r=1}^n \sum_{k=1}^r p_{S^*(k)} \) and
\[
\rho_3 = \frac{\sum T_j(S) + nd_{\text{max}}}{\sum T_j(S^*) + nd_{\text{max}}} \leq 1 + \frac{\sum T_j(S) - \sum T_j(S^*)}{\sum T_j(S^*) + nd_{\text{max}}}
\]

\[
\leq \frac{\sum_{r=1}^{n} \sum_{k=1}^{r} p_r}{\sum_{r=1}^{n} \sum_{k=1}^{r} p_r q_r^{a_1} q_r^{a_2} (1 + \beta_n \sum_{l=1}^{n} p_l)^a_2}
\]

\[
< \frac{1}{q_r^{a_1} (1 + \beta_n \sum_{l=1}^{n} p_l)^a_2}.
\]

It is not difficult to see that \(\rho_3 \to 1\) if the learning effect does not exist, i.e., \(a_1 = a_2 = 0\). It is intuitive because the EDD rule is optimal. \(\square\)

For \(1 | p_{j[r]} = p_j q_r^{a_1} (1 + \sum_{l=1}^{r-1} \beta_l p_l)^a_2 | \sum T_j\) problem, the EDD rule can not provide an optimal schedule. However, applying the similar technique of the proof in Theorem 3.1 and 3.3, the following results can be obtained.

**Theorem 3.8.** For the \(1 | p_{j[r]} = p_j q_r^{a_1} (1 + \sum_{l=1}^{r-1} \beta_l p_l)^a_2 | \sum T_j\) problem, an optimal schedule is obtained by sequencing jobs in non-decreasing order of \(d_j\), if the processing time and the due date of the jobs exist an agreeable relation, i.e., \(p_i \leq p_j \Rightarrow d_i \leq d_j\).

4. Two-machine flowshop problems which ordered or proportional job processing times. For convenience, we will write \(p_{j[r]} = p_j q_r^{a_1} (1 + \sum_{l=1}^{r-1} \beta_l p_l)^a_2\) for \(LE\). This section considers the four following flowshop problems:

1) \(F2|LE, \text{ord}|C_{\text{max}}\) and \(F2|LE, \text{ord}|\sum C_j\). Assume that \(a_j \leq b_j\) for job \(j\), if \(a_j \leq a_k\) for any two jobs \(j, k\), then \(b_j \leq b_k\);

2) \(F2|LE, \text{prp}|C_{\text{max}}\) and \(F2|LE, \text{prp}|\sum C_j\), with \(b_j = ca_j\) for each job \(j\) where \(c \geq 1\) is a constant factor, and \(a_j, b_j\) denote the length of the first and second operation of job \(j\) without any learning effects, respectively. Assume that both operations have the same learning rate, i.e.,

\[
a_{j[r]} = a_j q_r^{a_1} (1 + \sum_{l=1}^{r-1} \beta_l p_l)^a_2,
\]

\[
b_{j[r]} = b_j q_r^{a_1} (1 + \sum_{l=1}^{r-1} \beta_l p_l)^a_2.
\]

**Theorem 4.1.** For the two problems \(F2|LE, \text{ord}|C_{\text{max}}\) and \(F2|LE, \text{prp}|C_{\text{max}}\), the optimal sequence can be obtained by the SPT rule.

**Proof.** Assumed that there are two adjacent jobs \(i\) and \(k\) in the optimal sequence \(S^*\), job \(k\) sequenced in the \(r\)th position immediately preceding job \(i\) sequenced in the \((r + 1)\)th position, \(a_i \leq a_k\) and \(b_i \leq b_k\). Next, we will show that interchange of jobs \(i\) and \(k\) (generate a new job sequence \(S\)) does not increase the makespan, i.e., \(C_{\text{max}}(S) \leq C_{\text{max}}(S^*)\); repeated implementation of this argument will generate the optimality of the SPT sequence for the both the \(F2|LE, \text{ord}|C_{\text{max}}\) and \(F2|LE, \text{prp}|C_{\text{max}}\) problems.

Consider the \(F2|LE, \text{ord}|C_{\text{max}}\) problem. We can express the makespan under the job sequences \(S^*\) and \(S\) as follows

\[
C_{\text{max}}(S^*) = \sum_{l=1}^{n} b_{j_l[l]} + \sum_{l=1}^{n} I_{j_l[l]}, C_{\text{max}}(S) = \sum_{l=1}^{n} b_{j'_l[l]} + \sum_{l=1}^{n} I_{j'_l[l]}, \tag{10}
\]
where \((j_1, \cdots, j_n)\) and \((j'_1, \cdots, j'_n)\) are the job sequence under the schedules \(S^*\) and \(S\), respectively, and \(I_{j}[t]\) (respectively \(I'_{j}[t]\)) denotes the idle time occurring on the second machine immediately, prior to the processing of job \(j_{i}\) (respectively \(j'_{i}\)) in the sequence \(S^*\) (respectively, \(j'_{i}\) in the sequence \(S\)). Let \(C'_{r+1}\) and \(C''_{r+1}\) denote the second machine completion time of job \(i\) in the \((r+1)^{th}\) position in the sequences \(S^*\) and \(S\). Then the formulas in (10) specialize to the formulas \(C'_{r+1} = \sum_{l=1}^{r+1} b_{j_l'[t]} + \sum_{l=1}^{r+1} I_{j_l}[t]\),

\[C'_{r+1} = \sum_{l=1}^{r+1} b_{j_l'[t]} + \sum_{l=1}^{r+1} I_{j_l}[t].\]

From Theorem 3.2, we have \(\sum_{l=1}^{r+1} b_{j_l'[t]} \leq \sum_{l=1}^{r+1} b_{j_l}[t]\). Then,

\[C'_{r+1} \leq C'_{r+1}\] will hold if we can show that \(\sum_{l=1}^{r+1} I_{j_l}[t] \leq \sum_{l=1}^{r+1} I_{j_l}[t]\). The sums of the first \(q\) \((q = 1, 2, \cdots, n)\) idle times in the job sequences \(S^*\) and \(S\) are expressed as follows:

\[
\begin{align*}
\sum_{l=1}^{r+1} I_{j_l}[t] &= \max_{1 \leq j \leq q} \{Y_{j}[t]\}, \\
\sum_{l=1}^{r+1} I_{j_l}[t] &= \max_{1 \leq j \leq q} \{Y_{j}[t]\},
\end{align*}
\]

where \(Y_{j}[t] = \sum_{k=1}^{j} a_{j_k}[t] - \sum_{l=1}^{j-1} b_{j_l}[t], j = 1, 2, \cdots, n\). That is,

\[
\begin{align*}
\max_{1 \leq j \leq r+1} \{Y_{j}[t]\} &= \max_{1 \leq j \leq r+1} \left\{ \sum_{l=1}^{j} a_{j_l}[t] - \sum_{l=1}^{j-1} b_{j_l}[t] \right\} \\
&\leq \max_{1 \leq j \leq r+1} \{Y_{j}[t]\} = \max_{1 \leq j \leq r+1} \left\{ \sum_{l=1}^{j} a_{j_l}[t] - \sum_{l=1}^{j-1} b_{j_l}[t] \right\}.
\end{align*}
\]

Since the job sequences \((j_1, \cdots, j_n)\) and \((j'_1, \cdots, j'_n)\) under the schedules \(S^*\) and \(S\) differ only in the positions of job \(i\) and \(k\), which occupy the \(r\)th positions and the \((r+1)^{th}\) position, then \(Y_{j}[t] = Y'_{j}[t]\) for \(j = 1, 2, \cdots, r-1\). Next, \(\max\{Y_{r}[t], Y'_{r+1}[t]\} \leq Y_{r}[t]\) will be proved.

Let \(a_{t_0} = \sum_{l=1}^{r-1} a_{j_l}[t], a_{t_0}' = \sum_{l=1}^{r-1} a_{j_l'[t]}, b_{t_0} = \sum_{l=1}^{r-1} b_{j_l}[t], b_{t_0}' = \sum_{l=1}^{r-1} b_{j_l'[t]}\). Then we have

\[
\begin{align*}
Y'_{r}[t] &= a_{t_0}' + a_{i}[r] - b_{t_0}', \\
Y'_{r+1}[t] &= a_{t_0} + a_{i}[r] + a_{k}[r+1] - b_{t_0} - b_{i}[r], \\
Y_{r}[t] &= a_{t_0} + a_{k}[r] - b_{t_0}.
\end{align*}
\]

1. Consider the problem \(F^2|\text{\textit{L}E}, \text{\textit{ord}}|C_{\text{\textit{max}}}\). Note that \(a_{t_0} = a_{t_0}'\) and \(b_{t_0} = b_{t_0}'\), since the first \(r-1\) jobs are scheduled in an identical manner in job sequences \(S^*\) and \(S\). Based on \(a_i \leq a_k\), then \(a_{i}[r] \leq a_{k}[r]\). We can also obtain

\[
\begin{align*}
Y'_{r}[t] - Y_{r}[t] &= a_{t_0}' + a_{i}[r] - b_{t_0}' - (a_{t_0}' + a_{k}[r] - b_{t_0}) \\
&= a_{i}[r] - a_{k}[r] \\
&\leq 0.
\end{align*}
\]

Similarly, \(a_i \leq b_k\) can yield \(a_{i}[r] \leq b_{i}[r]\). Since \(r > 0\) and \(a \leq 0\), then \(a_{k}[r+1] \leq a_{k}[r]\). We have

\[
\begin{align*}
Y'_{r+1}[t] - Y_{r}[t] &= (a_{k}[r+1] - a_{k}[r]) + (a_{i}[r] - b_{i}[r]) \\
&\leq 0.
\end{align*}
\]
Theorem 4.2. For the two problems $F_2|LE, ord|\sum C_j$ and $F_2|LE, prp|\sum C_j$, the optimal sequence can be obtained by SPT rule.

Proof. Similar to Theorem 4.1. We will show that the interchange of job $i$ and job $k$ does not increase the total completion time. Let $C_k$ and $C_i$ (respectively $C'_k$ and $C'_i$) denote the completion times of job $k$ and job $i$ on the second machine before (after) the jobs interchange, respectively, i.e., $C_k + C_i \geq C'_k + C'_i$. Note that the interchange of job $i$ and job $k$ yields $C'_i \leq C_i$. Meanwhile, the completion times of any jobs after job $i$ and job $k$ can not change in the sequence. Consider that $a_{i[r]} \leq a_{k[r]}$ and $b_{i[r]} \leq b_{k[r]}$, then $C'_i \leq C_k$. This completes the proof. \qed

5. Conclusion. This paper provides scheduling models with both position-dependent learning and sum-of-processing-times learning phenomenon on single machine and flowshop machines. Our objective functions are to minimize the makespan, the total completion time, the total weighted completion time, the maximum lateness and the total tardiness on a single machine, and the makespan and the total completion times on the flowshop machines. We presented polynomial time algorithms to minimize the makespan and the total completion time on a single machine. We analyzed their worst case error bounds for minimizing the weighted completion time, the maximum lateness and the total tardiness by some class heuristic rules on a single-machine. Furthermore, we showed that these heuristic rules do provide optimal schedules under certain conditions. Finally we showed that minimizing the makespan problem and the total completion time problem can be solved by SPT rule under ordered processing times and proportional processing times.

It is suggested for future research to investigate the effects of learning in the context of the other scheduling problems, including multi-machine and job-shop settings. It may also consider a similar problem with more complex job learning functions.

REFERENCES


Received August 2017; 1st revision March 2018; 2nd revision March 2018.

E-mail address: zxg7980@163.com