Neural Comput & Applic (2016) 27:937–943
DOI 10.1007/s00521-015-1910-3

ORIGINAL ARTICLE

Single-machine scheduling with truncated sum-of-processing-times-based learning effect including proportional delivery times

Yu-Bin Wu[1] · Jian-Jun Wang[2]

Received: 4 August 2014 / Accepted: 7 April 2015 / Published online: 3 May 2015

Abstract The single-machine scheduling problem with truncated sum-of-processing-times-based learning effect and past-sequence-dependent job delivery times is considered. Each job's delivery time depends on its waiting time of processing. For some regular objective functions, it is proved that the problems can be solved by the smallest processing time first rule. For some special cases of the total weighted completion time and the maximum lateness objective functions, the thesis shows that the problems can be solved in polynomial time.

Keywords Scheduling · Delivery times · Learning effect · Single machine

1 Introduction

In traditional scheduling problems, job processing times are assumed as having fixed and constant values. However, in many real-life situations, it is not scarce that the actual processing time of a job is shorter if it is scheduled in the later sequence (for example, the machines and workers' performance can be improved by repeating the production operations), and this phenomenon is known as the "learning effect" [1]. An extensive survey of scheduling problems concerning learning effects is provided by Biskup [2]. More recently, Cheng et al. [3] considered scheduling problems with deteriorating jobs and learning effects, and the setup times are assumed to be proportional to the actual processing times of the already scheduled jobs. For some single-machine problems, they derived polynomial-time optimal solutions. Cheng et al. [4] considered a two-agent scheduling with truncated sum-of-prossing-times-based learning effect on a single machine. Wang and Wang [25, 26], Sun et al. [21, 22] and Wang et al. [33] considered flow-shop scheduling with learning effects. Wang et al. [23] considered two-machine makespan minimization a flow-shop scheduling with effects of deterioration and learning. Wang et al. [31] considered single-machine scheduling with a time-dependent learning effect and deteriorating jobs. They found some results to the total completion time minimization problem. Wang et al. [32] considered single-machine scheduling problems with truncated exponential learning effect, and they proved that some regular objective functions can be solved in polynomial time. Wu et al. [35] considered scheduling problems jobs with a truncated sum-of-processing-times-based learning effect, and they proved that some single-machine scheduling problems can be solved in polynomial time. Wu et al. [36] considered scheduling problems jobs with a truncated position-based learning effect, and they proved that some single-machine and two-machine flow-shop scheduling problems can be solved in polynomial time. Other types of scheduling problems and models with learning effects have also been discussed, and more references can be found in papers by Zhang et al. [44], Rudek [18, 19], Cheng et al. [5, 6], Wang et al. [24, 32], Lu et al. [16], Wang and Wang [27–30], Yang et al. [37], Niu et al. [17], and Li et al. [10].

✉ Yu-Bin Wu
wuyubin79@163.com

Jian-Jun Wang
drwangjj@dlut.edu.cn

1 School of Science, Shenyang Aerospace University, Shenyang 110136, China

2 Faculty of Management and Economics, Dalian University of Technology, Dalian 116024, China

Koulamas and Kyparisis [9] and Liu et al. [14, 15] considered scheduling with past-sequence-dependent (p-s-d) job delivery times on a single machine. For some regular and non-regular objective functions, Koulamas and Kyparisis [9] and Liu et al. [15] showed that the problem can be solved by simple polynomial-time algorithms. Liu et al. [14] considered the problem with p-s-d job delivery times and the release times. Liu et al. [13] and Yin et al. [41] studied scheduling with p-s-d job delivery times and deteriorating jobs. However, to the best of our knowledge, apart from the recent paper of Yang et al. [38, 40], Yang and Yang [39], Liu [12], Yin et al. [42], Shen and Wu [20], and Zhao and Tang [43], it has not been investigated in the scheduling models by considering learning effect and the p-s-d delivery time simultaneously. Yang et al. [38] studied scheduling with (p-s-d) job delivery times and job-independent learning effect (i.e., the actual processing time of job J_j if it is scheduled in position r on a single machine is given by: $p^A_{jr} = p_j r^a$, where $a \le 0$ is a constant learning effect. Yang and Yang [39] studied scheduling with p-s-d job delivery times and position-dependent processing times (i.e., the actual processing time of job J_j if it is scheduled in position r on a single machine is given by: $p^A_{jr} = p_j r^{a_j}$, where a_j is a job-dependent factor. Liu [12] considered the identical parallel-machine scheduling problem with p-s-d and job-independent learning effect (the same as in Yang et al. [38]). For the total absolute deviation of job completion times minimization, the total load on all machines minimization and the total completion time minimization, they proposed polynomial algorithms to optimally solve these problems. Yin et al. [42] considered the single-machine scheduling problem with p-s-d and job-independent learning effect (the same as in Yang et al. [38]). For four due date determination decisions, they proved that the proposed problems can be solved in polynomial time. Yang et al. [40] considered p-s-d delivery times scheduling problems with deterioration and learning effects simultaneously on a single machine. Shen and Wu [20] considered single-machine scheduling with p-s-d job delivery times and general position-dependent and time-dependent learning effects (i.e., the actual processing time of job J_j if it is scheduled in position r is given by: $p^A_{jr} = p_j f(\sum_{i=1}^{r-1} p_{[i]}) g(r)$, where f is a differentiable non-increasing function with f' non-decreasing on $[0, +\infty)$ and $f(0) = 1$ and g is a non-increasing function with $g(1) = 1$. Under the proposed learning model, they provided optimal solutions for some regular objective functions. Zhao and Tang [43] considered single-machine scheduling with p-s-d job delivery times and general position-dependent processing times (i.e., the actual processing time of job J_j if it is scheduled in position r is given by: $p^A_{jr} = p_j g_j(r)$, where $g_j(r)$ is a function that specifies a job-dependent positional effect. Under the proposed learning model, they provided optimal solutions for some objective functions.

However, subjected to the uncontrolled learning effect, the actual processing time of a job will plummet to zero dramatically due to the increasing number of jobs which are already processed or the emergence of jobs with long processing time. Hence, Cheng et al. [4], Wu et al. [35, 36], Wang et al. [24] and Li et al. [11] considered the truncated learning effect model. Hence, in this paper we extend the results of Yang et al. [38, 40], Yang and Yang [39], Liu [12], Yin et al. [42], Shen and Wu [20], and Zhao and Tang [43] to the single-machine scheduling problem with the truncated sum-of-processing-times-based learning effect and p-s-d delivery times. The phenomenon of the truncated sum-of-processing-times-based learning effect and past-sequence-dependent (p-s-d) job delivery times can be found in a production situation, "for example, an electronic component may be exposed to certain electromagnetic and/or radioactive fields while waiting in the machine's preprocessing area, and regulatory authorities require the component to be 'treated' (e.g., in a chemical solution capable of removing/neutralizing certain effects of electromagnetic/radioactive fields) for an amount of time proportional to the job's exposure time to these fields. This treatment can be performed after the component has been processed by the machine, but before it is delivered to the customer so it can be delivered with a 'guarantee.' Such a postprocessing operation is usually called the job 'delivery time,' i.e., the past-sequence-dependent (p-s-d) job delivery times. In addition, if human interactions (the machines or workers) have a significant impact during the processing of the job, the processing time will add to the workers experience and cause an uncontrolled learning effect" [9, 20].

The rest part of the paper is organized as follows: Sect. 2 formulates the problem. Section 3 considers several single-machine scheduling. Section 4 provides some extensions. In Sect. 5, conclusions are drawn.

2 Problems description

The notations used throughout this paper are defined as follows.

n :	The total number of jobs
p_j :	The normal processing time of job J_j, $j = 1, 2, \ldots, n$. The normal processing time of a job is incurred when the job is scheduled first in a sequence
v_j:	The weight of job J_j, $j = 1, 2, \ldots, n$
d_j:	The due date of job J_j, $j = 1, 2, \ldots, n$

$p_{[r]}$:	The normal processing time of a job scheduled in the rth position in a sequence, $r = 1, 2, \ldots, n$
$q_{[r]}$:	The past-sequence-dependent (p-s-d) delivery time of a job scheduled in the rth position in a sequence, $r = 1, 2, \ldots, n$
$w_{[r]}$:	The waiting time of a job scheduled in the rth position in a sequence, $r = 1, 2, \ldots, n$
p^A_{jr} :	The actual processing time of job J_j scheduled in the rth position in a sequence, $r, j = 1, 2, \ldots, n$
$p^A_{[r]}$:	Be the actual processing time of a job if it is scheduled in the rth position in a sequence, $r = 1, 2, \ldots, n$
π :	A job sequence of n jobs
$C_j = C_j(\pi)$:	The completion time for job J_j in π
$C_{\max}$:	The makespan of a given permutation, i.e., $C_{\max} = \max\{C_j \mid j = 1, 2, \ldots, n\}$
$\sum C_j$:	The sum of completion times of jobs
$\sum C_j^\theta$:	The sum of the θth ($\theta \geq 0$) power of job completion times
$\sum v_j C_j$:	The total weighted completion time
$\sum L_j$:	The total lateness
$L_{\max} = \max\{C_j - d_j \mid j = 1, 2, \ldots, n\}$:	The maximum lateness

The model is described as follows. There are n independent and non-preemptive jobs $J = \{J_1, J_2, \ldots, J_n\}$ to be processed on a single machine. All the jobs are available at time 0; the machine will handle one job at a time, and there will be no idle time between the processes of any job. As in Cheng et al. [4], Wu et al. [35] and Li et al. [11], we assume that the actual processing time of job J_j, scheduled in position r, is given by:

$$p^A_{jr} = p_j \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a, \beta\right\}, \quad r, j = 1, 2, \ldots, n, \tag{1}$$

where $a \leq 0$ is a constant learning effect, $p_{[0]} = 0$, $\sum_{i=1}^{0} p_{[i]} = 0$ and β is a truncation parameter with $0 < \beta < 1$. Moreover, following Koulamas and Kyparisis [9], Yang et al. [38], Yang and Yang [39] and Shen and Wu [20], the processing of the job $J_{[r]}$ must be followed by the p-s-d delivery time $q_{[r]}$, i.e.,

$$q_{[1]} = 0 \quad \text{and} \quad q_{[r]} = \gamma w_{[r]} = \gamma \sum_{i=1}^{r-1} p^A_{[i]}, \quad r = 2, 3, \ldots n, \tag{2}$$

where $\gamma \geq 0$ is a normalizing constant and $w_{[r]}$ denotes the waiting time of job $J_{[r]}$. Let $C_{[j]}$ denote the completion time of job $J_{[j]}$ and C_j is defined analogously. We assume that the postprocessing operation of any job $J_{[j]}$ modeled by its delivery time $q_{[j]}$ is performed "off-line," i.e., $C_{[1]} = p_{[1]}$ and

$$C_{[j]} = w_{[j]} + p^A_{[j]} + q_{[j]}, \quad j = 2, 3, \ldots, n,$$

where $w_{[1]} = 0$ and $w_{[j]} = \sum_{i=1}^{j-1} p^A_{[i]}$, $j = 2, 3, \ldots, n$.

For convenience, we denote by $TTLE$ the truncated time-dependent learning effect given by (1) and denote by q_{psd} the p-s-d delivery given by (2) [9]. In this paper, we consider the problem to minimize the makespan $C_{\max} = \max\{C_j \mid j = 1, 2, \ldots, n\}$, the total completion time $\sum C_j$, the sum of the θth ($\theta \geq 0$) power of job completion times $\sum C_j^\theta$, the total lateness $\sum L_j$, the total weighted completion time $\sum v_j C_j$ and the maximum lateness $L_{\max} = \max\{C_j - d_j \mid j = 1, 2, \ldots, n\}$, where v_j and d_j denote the weight and due date of job J_j. Using the three-field notation of Graham et al. [8], we denote the problems that are considered in this paper as $1|TTLE, q_{psd}|\gamma$, where $\gamma \in \{C_{\max}, \sum C_j^\theta, \sum L_j, \sum v_j C_j, L_{\max}\}$.

3 Single-machine scheduling problems

Lemma 1 *For the $1|TTLE|C_{\max}$ problem, an optimal schedule can be obtained by the SPT rule (i.e., by sequencing the jobs in non-decreasing order of p_j).*

Proof See the proof in Wu et al. [35]. □

Theorem 1 *For the $1|TTLE, q_{psd}|C_{\max}$ problem, an optimal schedule can be obtained by the SPT rule (i.e., by sequencing the jobs in non-decreasing order of p_j).*

Proof Let $\pi = [S_1, J_j, J_k, S_2]$ and $\pi' = [S_1, J_k, J_j, S_2]$ be two job schedules by interchanging two adjacent jobs J_j and J_k, where S_1 and S_2 are partial sequences and S_1 or S_2 may be empty. We also assume that there are $r-1$ jobs 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 $C_k(\pi) \leq C_j(\pi')$ (since for any J_u in S_2, from $C_k(\pi) \leq C_j(\pi')$ we have $C_u(\pi) \leq C_u(\pi')$, hence $C_{[n]}(\pi) \leq C_{[n]}(\pi')$). Under π, the completion times of jobs J_j and J_k are

$$C_j(\pi) = \sum_{i=1}^{r-1} p_{[i]}^A + p_j \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a, \beta\right\} + \gamma \sum_{i=1}^{r-1} p_{[i]}^A.$$

$$\begin{aligned} C_k(\pi) = & \sum_{i=1}^{r-1} p_{[i]}^A + p_j \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a, \beta\right\} \\ & + p_k \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]} + p_j\right)^a, \beta\right\} \\ & + \gamma\left[\sum_{i=1}^{r-1} p_{[i]}^A + p_j \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a, \beta\right\}\right], \end{aligned} \tag{3}$$

whereas under π', they are

$$C_k(\pi') = \sum_{i=1}^{r-1} p_{[i]}^A + p_k \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a, \beta\right\} + \gamma \sum_{i=1}^{r-1} p_{[i]}^A$$

and

$$\begin{aligned} C_j(\pi') = & \sum_{i=1}^{r-1} p_{[i]}^A + p_k \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a, \beta\right\} \\ & + p_j \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]} + p_k\right)^a, \beta\right\} \\ & + \gamma\left[\sum_{i=1}^{r-1} p_{[i]}^A + p_k \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a, \beta\right\}\right]. \end{aligned} \tag{4}$$

Stem from Eqs. (3) and (4), we have

$$\begin{aligned} C_j(\pi') - C_k(\pi) = & \gamma\ (p_k - p_j) \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a, \beta\right\} \\ & + p_k \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a, \beta\right\} \\ & + p_j \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]} + p_k\right)^a, \beta\right\} \\ & - p_j \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a, \beta\right\} \\ & - p_k \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]} + p_j\right)^a, \beta\right\}. \end{aligned}$$

From $p_j \leq p_k$ and Lemma 1, we have $\gamma\ (p_k - p_j)\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a \geq 0$ and $p_k \max\left\{(1 + \sum_{i=1}^{r-1} p_{[i]})^a, \beta\right\} + p_j \max\left\{(1 + \sum_{i=1}^{r-1} p_{[i]} + p_k)^a, \beta\right\} - p_j \max\left\{(1 + \sum_{i=1}^{r-1} p_{[i]})^a, \beta\right\} - p_k \max\left\{(1 + \sum_{i=1}^{r-1} p_{[i]} + p_j)^a, \beta\right\} \geq 0$, hence $C_j(\pi') - C_k(\pi) \geq 0$. □

Theorem 2 *For the problem $1|TTLE, q_{psd}|\sum C_j^\theta$, an optimal schedule can be obtained by the SPT rule.*

Proof We also use the same notations as in Theorem 1. In order to show π dominates π', we should prove that (i) $C_k(\pi) \leq C_j(\pi')$ and (ii) $C_j^\theta(\pi) + C_k^\theta(\pi) \leq C_k^\theta(\pi') + C_j^\theta(\pi')$ (since for any J_u in S_2, from $C_k(\pi) \leq C_j(\pi')$ we have $C_u(\pi) \leq C_u(\pi')$, hence $\sum C_j^\theta(\pi) \leq \sum C_j^\theta(\pi')$).

Case (i) has been proved by Theorem 1. From $p_j \leq p_k$, we have $C_j(\pi) \leq C_k(\pi')$, hence

$$C_j^\theta(\pi) + C_k^\theta(\pi) \leq C_k^\theta(\pi') + C_j^\theta(\pi').$$

It completes the proof of case (ii). □

Corollary 1 *For the $1|TTLE, q_{psd}|\sum C_j$ problem, an optimal schedule can be obtained by the SPT rule.*

Theorem 3 *For the $1|TTLE, q_{psd}|\sum L_j$ problem, an optimal schedule can be obtained by the SPT rule.*

Proof Obviously,

$$\sum_{j=1}^{n} L_j = \sum_{j=1}^{n} (C_j - d_j) = \sum_{j=1}^{n} C_j - \sum_{j=1}^{n} d_j.$$

For $\sum_{j=1}^{n} d_j$ is a constant, $\sum_{j=1}^{n} L_j$ is minimized if $\sum_{j=1}^{n} C_j$ is minimized. From Theorem 2, the result can be easily obtained. □

Theorem 4 *For the $1|TTLE, q_{psd}|\sum v_j C_j$ problem, if all the jobs have agreeable weights, i.e., $p_j \leq p_k$ implies $v_j \geq v_k$ for all the jobs J_j and J_k, an optimal schedule can be obtained by the WSPT rule (i.e., by sequencing the jobs in non-decreasing order of p_j/v_j).*

Proof As in Theorem 1, the same notations are also used. In order to show π dominates π', we should prove that (1) $C_k(\pi) \leq C_j(\pi')$ and (2) $v_j C_j(\pi) + v_k C_k(\pi) \leq v_k C_k(\pi') + v_j C_j(\pi')$ with $p_j/v_j \leq p_k/v_k$, $p_j \leq p_k$ and $v_j \geq v_k$.

Case (1) has been proved by Theorem 1. We provide the proof of case (2) as follows.

From Theorem 1, we have $C_j(\pi) \leq C_k(\pi')$, hence

$$\begin{aligned} & v_k C_k(\pi') + v_j C_j(\pi') - v_j C_j(\pi) - v_k C_k(\pi) \\ & \geq v_k C_j(\pi) + v_j C_k(\pi) - v_j C_j(\pi) - v_k C_k(\pi) \\ & = (v_j - v_k)(C_k(\pi) - C_j(\pi)) \\ & \geq 0. \end{aligned}$$

Hence, $v_j C_j(\pi) + v_k C_k(\pi) \leq v_k C_k(\pi') + v_j C_j(\pi')$. It completes the proof of part (2) and thus the theorem. □

Theorem 5 *For the problem $1|TTLE, q_{psd}|L_{\max}$, if the jobs have agreeable conditions, i.e., $p_j \leq p_k$ implies $d_j \leq d_k$ for all the jobs J_j and J_k, an optimal schedule can be obtained by the EDD rule (i.e., by sequencing the jobs in non-decreasing order of d_j).*

Proof We still use the same notations mentioned above. From the proof of Theorem 1, under π, the lateness of jobs J_j and J_k are

$$L_j(\pi) = \sum_{i=1}^{r-1} p_{[i]}^A + p_j \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a, \beta\right\} + \gamma \sum_{i=1}^{r-1} p_{[i]}^A - d_j,$$

$$L_k(\pi) = \sum_{i=1}^{r-1} p_{[i]}^A + p_j \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a, \beta\right\} + p_k \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]} + p_j\right)^a, \beta\right\} + \gamma\left[\sum_{i=1}^{r-1} p_{[i]}^A + p_j\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a\right] - d_k,$$

whereas under π', they are

$$L_k(\pi') = \sum_{i=1}^{r-1} p_{[i]}^A + p_k \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a, \beta\right\} + \gamma \sum_{i=1}^{r-1} p_{[i]}^A - d_k,$$

$$L_j(\pi') = \sum_{i=1}^{r-1} p_{[i]}^A + p_k \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]}\right)^a, \beta\right\} + p_j \max\left\{\left(1 + \sum_{i=1}^{r-1} p_{[i]} + p_k\right)^a, \beta\right\} + \gamma\left[\sum_{i=1}^{r-1} p_{[i]}^A + p_k(1 + \sum_{i=1}^{r-1} p_{[i]})^a\right] - d_j.$$

If $d_j \le d_k$, we have $L_k(\pi') \le L_j(\pi')$. In addition, if $d_j \le d_k$ and $p_j \le p_k$, according to Theorem 1, we get $L_k(\pi) \le L_j(\pi')$ and $L_j(\pi) \le L_j(\pi')$. Therefore, we obtain

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

It completes the proof of the theorem. □

The following example illustrates the results of the theorems.

Example 1 Consider the instance with $n = 3$, $p_1 = 10$, $p_2 = 12$, $p_3 = 14$, $v_1 = 8$, $v_2 = 5$, $v_3 = 3$, $d_1 = 11$, $d_2 = 14$, $d_3 = 16$, $a = -0.3$, $\beta = 0.8$, $\gamma = 0.5$ and $\theta = 2$. Now applying Theorems 1–5, we know that the optimal schedule is $[J_1, J_2, J_3]$ for the following objective functions: $C_{\max}$, $\sum C_j^2$, $\sum L_j$, $\sum v_j C_j$ and $L_{\max}$. In addition, we have

$$C_1 = 10, C_2 = 10 + 12 * \max\left\{(1 + 10)^{-0.3}, 0.8\right\} + 0.5 * 10 = 24.6,$$

$$C_3 = 24.6 + 14 * \max\left\{(1 + 10 + 12)^{-0.3}, 0.8\right\} + 0.5 * 24.6 = 48.1.$$

$C_{\max} = C_3 = 48.1$,
$\sum C_j^2 = 10^2 + 24.6^2 + 48.1^2 = 3018.8$,
$\sum L_j = 10 + 24.6 + 48.1 - 11 - 14 - 16 = 41.7$,
$\sum v_j C_j = 8 * 10 + 5 * 24.6 + 3 * 48.1 = 347.3$ and
$L_{\max} = \max\{10 - 11, 24.6 - 14, 48.1 - 16\} = 32.1$.

4 Extensions

Similar to the proof of Sect. 3 and Shen and Wu [20], our results (Theorems 1–5) can be extended to the following learning effect models:

(a) The model of Zhang et al. [44]:

$$p_{jr}^A = p_j\left(q_r + \sum_{l=1}^{r-1} \beta_l \ln p_{[l]}\right)^a, \tag{5}$$

where a is the learning indices with $a \le 0$, β_r is the weight of position r, $0 \le \beta_1 \le \beta_2 \le \cdots \le \beta_n$ and $0 \le q_1 \le q_2 \le \cdots \le q_n$.

(b) The model of Rudek [18]:

$$p_{jr}^A = p_j f\left(\sum_{l=1}^{r-1} p_{[l]}\right), \tag{6}$$

where $f : [1, +\infty) \to (0, 1]$ is the non-increasing function (i.e., learning curve) common for all jobs.

(c) The model of Cheng et al. [5]:

$$p_{jr}^A = p_j\left(1 + \sum_{l=1}^{r-1} \beta_l \ln p_{[l]}\right)^a r^b, \tag{7}$$

where $\sum_{l=1}^{0} \beta_l \ln p_{[l]} := 0$, a and b are the learning indices with $a \le 0$ and $b \le 0$, and β_r is the weight of position r.

(d) The model of Wang et al. [32]:

$$p_{jr}^A = p_j \max\left\{\left(\mu a^{-\frac{\sum_{i=1}^{r-1} p_{[i]}}{\sum_{i=1}^{n} p_{[i]}}} + v\right), \beta\right\}, \quad r, j = 1, 2, \ldots, n, \tag{8}$$

where $\mu \le 0$, $v \le 0$, $\mu + v = 1$, $\sum_{i=1}^{0} p_{[i]} = 0$ and β is a truncation parameter with $0 < \beta < 1$.

(e) The model of Cheng et al. [6]:

$$p_{jr}^A = p_j\left(1 + \sum_{l=1}^{r-1} \varphi_l p_{[l]}\right)^a r^b, \tag{9}$$

where $a \le 0$ and $b \le 0$ and $\varphi_l > 0$ is the associated weight of the lth position.

(f) The model of Niu et al. [17]:

$$p_{jr}^A = p_j f\left(\sum_{l=1}^{r-1} g(p_{[l]})\right) h(r), \tag{10}$$

where $\frac{df(x)}{dx} \leq 0, \frac{d^2f(x)}{dx^2} \geq 0, \frac{d^2g(x)}{dx^2} \leq 0$ and $h(r)$ is a non-increasing function with $g(0) = 0$ and $h(1) = 1$.

5 Conclusions

This research studied the scheduling problems with truncated sum-of-processing-times-based learning effect and past-sequence-dependent job delivery times on a single machine. To some single-machine minimization problems, it is proved that the proposed problems were polynomial time solvable. In future research, more realistic settings such as the job-shop scheduling [7, 34] and unrelated parallel machines scheduling, or optimizing other performance measures with truncated sum-of-processing-times-based learning effect and past-sequence-dependent job delivery times will be explored.

Acknowledgments We are grateful to the anonymous referees for their helpful comments on an earlier versions of this note. This research was supported by the Science Research Foundation of Shenyang Aerospace University (Grant No. 201304Y), the National Natural Science Foundation of China (Grant Nos. 71471120, 71271039), New Century Excellent Talents in University (NCET-13-0082), and Changjiang Scholars and Innovative Research Team in University (IRT1214).

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Badiru AB (1992) Computational survey of univariate and multivariate learning curve models. IEEE Trans Eng Manag 39:176–188
2. Biskup D (2008) A state-of-the-art review on scheduling with learning effects. Eur J Oper Res 188:315–329
3. Cheng TCE, Wu C-C, Lee W-C (2010) Scheduling problems with deteriorating jobs and learning effects including proportional setup times. Comput Ind Eng 58:326–331
4. Cheng TCE, Cheng S-R, Wu W-H, Hsu P-H, Wu C-C (2011) A two-agent single-machine scheduling problem with truncated sum-of-processing-times-based learning considerations. Comput Ind Eng 60:534–541
5. Cheng TCE, Kuo W-H, Yang D-L (2013) Scheduling with a position-weighted learning effect based on sum-of-logarithm-processing-times and job position. Inf Sci 221:490–500
6. Cheng TCE, Kuo W-H, Yang D-L (2014) Scheduling with a position-weighted learning effect. Optim Lett 8:293–306
7. Geyik F, Dosdoğru AT (2013) Process plan and part routing optimization in a dynamic flexible job shop scheduling environment: an optimization via simulation approach. Neural Comput Appl 23:1631–1641
8. Graham RL, Lawler EL, Lenstra JK, Rinnooy Kan AHG (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann Discret Math 5:287–326
9. Koulamas C, Kyparisis GJ (2010) Single-machine scheduling problems with past-sequence-dependent delivery times. Int J Prod Econ 126:264–266
10. Li G, Luo M-L, Zhang W-J, Wang X-Y (2015) Single-machine due-window assignment scheduling based on common flow allowance, learning effect and resource allocation. Int J Prod Res 53(4):1228–1241
11. Li L, Yang S-W, Wu Y-B, Huo Y, Ji P (2013) Single machine scheduling jobs with a truncated sum-of-processing-times-based learning effect. Int J Adv Manuf Technol 67:261–267
12. Liu M (2013) Parallel-machine scheduling with past-sequence-dependent delivery times and learning effect. Appl Math Model 37:9630–9633
13. Liu M, Wang S, Chu C (2013) Scheduling deteriorating jobs with past-sequence-dependent delivery times. Int J Prod Econ 144:418–421
14. Liu M, Zheng FF, Chu CB, Yf Xu (2012) Single-machine scheduling with past-sequence-dependent delivery times and release times. Inf Process Lett 112:835–838
15. Liu M, Zheng FF, Chu CB, Yf Xu (2012) New results on single-machine scheduling with past-sequence-dependent delivery times. Theor Comput Sci 438:55–61
16. Lu Y-Y, Li G, Wu Y-B, Ji P (2014) Optimal due-date assignment problem with learning effect and resource-dependent processing times. Optim Lett 8:113–127
17. Niu Y-P, Wan L, Wang J-B (2015) A note on scheduling jobs with extended sum-of-processing-times-based and position-based learning effect. Asia Pac J Oper Res 32(2):1550001 (12 pages)
18. Rudek R (2012) The single processor total weighted completion time scheduling problem with the sum-of-processing-time based learning model. Inf Sci 199:216–229
19. Rudek R (2013) On single processor scheduling problems with learning dependent on the number of processed jobs. Appl Math Model 37:1523–1536
20. Shen L, Wu Y-B (2013) Single machine past-sequence-dependent delivery times scheduling with general position-dependent and time-dependent learning effects. Appl Math Model 37:5444–5451
21. Sun L-H, Cui K, Chen J-H, Wang J, He X-C (2013) Some results of the worst-case analysis for flow shop scheduling with a learning effect. Ann Oper Res 211:481–490
22. Sun L-H, Cui K, Chen J-H, Wang J, He X-C (2013) Research on permutation flow shop scheduling problems with general position-dependent learning effects. Ann Oper Res 211:473–480
23. Wang J-B, Ji P, Cheng TCE, Wang D (2012) Minimizing makespan in a two-machine flow shop with effects of deterioration and learning. Optim Lett 6:1393–1409
24. Wang X-R, Jin J, Wang J-B, Ji P (2014) Single machine scheduling with truncated job-dependent learning effect. Optim Lett 8:669–677
25. Wang J-B, Wang M-Z (2011) Worst-case behavior of simple sequencing rules in flow shop scheduling with general position-dependent learning effects. Ann Oper Res 191:155–169
26. Wang J-B, Wang M-Z (2012) Worst-case analysis for flow shop scheduling problems with an exponential learning effect. J Oper Res Soc 63:130–137
27. Wang J-B, Wang M-Z (2014) Single-machine due-window assignment and scheduling with learning effect and resource-dependent processing times. Asia Pac J Oper Res 31(5):1450036 (15 pages)
28. Wang J-B, Wang J-J (2014) Single machine scheduling with sum-of-logarithm-processing-times based and position based learning effects. Optim Lett 8:971–982
29. Wang J-B, Wang J-J (2014) Flowshop scheduling with a general exponential learning effect. Comput Oper Res 43:292–308
30. Wang X-Y, Wang J-J (2014) Scheduling deteriorating jobs with a learning effect on unrelated parallel-machines. Appl Math Model 38:5231–5238
31. Wang J-B, Wang M-Z, Ji P (2012) Single machine total completion time minimization scheduling with a time-dependent learning effect and deteriorating jobs. Int J Syst Sci 43:861–868

32. Wang J-B, Wang X-Y, Sun L-H, Sun L-Y (2013) Scheduling jobs with truncated exponential learning functions. Optim Lett 7:1857–1873
33. Wang X-Y, Zhou Z, Zhang X, Ji P, Wang J-B (2013) Several flow shop scheduling problems with truncated position-based learning effect. Comput Oper Res 40:2906–2929
34. Weckman G, Bondal AA, Rinder MM, Young WA II (2012) Applying a hybrid artificial immune systems to the job shop scheduling problem. Neural Comput Appl 21:1465–1475
35. Wu C-C, Yin Y, Wu W-H, Cheng S-R (2012) Some polynomial solvable single-machine scheduling problems with a truncation sum-of-processing-times based learning effect. Eur J Ind Eng 6:441–453
36. Wu C-C, Yin Y, Cheng S-R (2013) Single-machine and two-machine flowshop scheduling problems with truncated position-based learning functions. J Oper Res Soc 64:147–156
37. Yang D-L, Cheng TCE, Kuo W-H (2013) Scheduling with a general learning effect. Int J Adv Manuf Technol 67:217–229
38. Yang S-J, Hsu C-J, Chang T-R, Yang D-L (2011) Single-machine scheduling with past-sequence-dependent delivery times and learning effect. J Chin Inst Ind Eng 28:247–255
39. Yang S-J, Yang D-L (2012) Single-machine scheduling problems with past-sequence-dependent delivery times and position-dependent processing times. J Oper Res Soc 63:1508–1515
40. Yang S-J, Guo J-Y, Lee H-T, Yang D-L (2013) Single-machine scheduling problems with past-sequence-dependent delivery times and deterioration and learning effects simultaneously. Int J Innov Comput Inf Control 9:3981–3989
41. Yin Y, Cheng TCE, Xu J, Cheng S-R, Wu C-C (2013) Single-machine scheduling with past-sequence-dependent delivery times and a linear deterioration. J Ind Manag Optim 9:323–339
42. Yin Y, Liu M, Cheng TCE, Wu C-C, Cheng S-R (2013) Four single-machine scheduling problems involving due date determination decisions. Inf Sci 251:164–181
43. Zhao C, Tang H (2014) Single machine scheduling problems with general position-dependent processing times and past-sequence-dependent delivery times. J Appl Math Comput 45:259–274
44. Zhang X, Yan G, Huang W, Tang G (2012) A note on machine scheduling with sum-of-logarithm-processing-time-based and position-based learning effects. Inf Sci 187:298–304