J Glob Optim
DOI 10.1007/s10898-015-0320-5

Serial-batching scheduling with time-dependent setup time and effects of deterioration and learning on a single-machine

Jun Pei[1,2] · Xinbao Liu[1,3] · Panos M. Pardalos[2] · Athanasios Migdalas[4,5] · ShanlinYang[1,3]

Received: 2 April 2015 / Accepted: 3 June 2015

Abstract This paper deals with serial-batching scheduling problems with the effects of deterioration and learning, where time-dependent setup time is also considered. In the proposed scheduling models, all jobs are first partitioned into serial batches, and then all batches are processed on a single serial-batching machine. The actual job processing time is a function of its starting time and position. In addition, a setup time is required when a new batch is processed, and the setup time of the batches is time-dependent, i.e., it is a linear function of its starting time. Structural properties are derived for the problems of minimizing the makespan, the number of tardy jobs, and the maximum earliness. Then, three optimization algorithms are developed to solve them, respectively.

Keywords Scheduling · Serial-batching · Deteriorating jobs · Learning effect · Single-machine

1 Introduction

In the majority of the scheduling literature, it was often assumed that the processing time of jobs is fixed. However, the processing times may change in many real-world situations

✉ Jun Pei
feiyijun.ufl@gmail.com

1 School of Management, Hefei University of Technology, Hefei, China

2 Department of Industrial and Systems Engineering, Center for Applied Optimization, University of Florida, Gainesville, FL, USA

3 Key Laboratory of Process Optimization and Intelligent Decision-Making, Ministry of Education, Hefei, China

4 Division of Industrial Logistics, Department of Industrial Engineering, Lulea University of Technology, Luleå, Sweden

5 Division of Transportation, Construction Management and Regional Planning, Department of Civil Engineering, Aristotle University of Thessaloniki, Thessaloníki, Greece

Published online: 07 June 2015

because of learning and/or deterioration effects. After repeating the same or similar operations, the machines or workers can improve the production efficiency and the learning effect occurs. Biskup [1] and Cheng and Wang [2] were the pioneers introducing the learning effect into scheduling problems. A position-dependent learning model was developed by Biskup [1]. Cheng and Wang introduced the model of learning effect using a volume-dependent processing time function. A lengthier survey was given by Mosheiov and Sidney [3]. On the other hand, job deterioration appears in many production situations. Gupta and Gupta [4] and Browne and Yechiali [5] initiated the research on scheduling with the deteriorating effect. Readers may refer to the extensive reviews on deteriorating jobs models by Cheng et al. [6], Gawiejnowicz [7], and Jafari and Moslehi [8].

The batch production, as an important class of scheduling problems, exists in many production situations. There are two main types of processing way of batching scheduling, referring to parallel- and serial-batching. In recent years some papers have addressed the parallel-batching scheduling problems with deterioration and/or learning effects, including Qi et al. [9], Yang and Kuo [10], Li et al. [11], and Miao et al. [12]. However, the research field on serial-scheduling problems with the effects of deterioration and learning is relatively unexplored. In our previous research [13], the problem of coordinated production and transportation was investigated, where the deteriorating jobs are processed on a serial-batching machine in an aluminum manufacturing factory. After further research on aluminum manufacturing process, we found that the workers and machines can increase the processing efficiency by repeating the production operations. That is, the effect of learning is also observed during the aluminum ingots processing. Then, the serial-batching scheduling problem with the effects of deterioration and learning is proposed in the present paper. In addition, setup time is required before processing each batch and the deterioration of setup time is undertaken during the production, i.e., it is time-dependent of its starting time. Thus, based on these two production characteristics, i.e., time-dependent setup times and the effects of deterioration and learning, the problem studied in the present paper is different from that in our previous work [13].

In today's manufacturing environment, scheduling with setup operations plays a crucial role. Setup time has been taken into account in many previous studies on scheduling problems with the effects of deterioration and/or learning. In [14,15], it was assumed that the setup time for a job is past-sequence-dependent and proportional to the actual processing times of the already scheduled jobs. Wu and Lee [16] studied single-machine group-scheduling problems with deteriorating setup times, where the actual setup time is a linear function of its starting time. Although setup time has been considered in batching scheduling problems [17,18], it was commonly assumed to be a constant, which may be inconsistent with real situations.

The group scheduling problems with the effect of deterioration or learning are similar to the problem studied in this paper. Based on similar production requirements, the jobs are classified into certain groups in advance in the group scheduling problems. Yang [19] studied the single-machine group scheduling problems with the effects of deterioration and learning at the same time. Three types of scheduling models on jobs deterioration were investigated, and polynomial time algorithms were developed for these models with the objective of minimizing makespan. In addition, the author proved that the total completion time minimization problem can be also solved in polynomial time under certain conditions. Bai et al. [20] investigated the group scheduling problem with the consideration of deterioration and learning effects on a single machine. The problems with the objectives of minimizing the makespan and the sum of completion time were considered, and polynomial time algorithms were developed to solve them, respectively. More recent papers that studied group scheduling problems

Table 1 Differences between group and serial-batching scheduling problems [13]

Factors	Group scheduling problems	Serial-batching scheduling problems
Classifying/batching	Jobs have been classified into certain groups beforehand	Both decisions on jobs batching and batches sequencing need to be made simultaneously
Job completion time	The jobs in the same group have different completion times	The jobs in the same batch have the same completion time which is equal to the completion time of the last job in that batch
Machine capacity	The machine capacity doesn't need to be considered	The machine capacity should be taken into account
Setup time	Setup time is needed before each group if it is considered	Setup time is required before each batch

with the effects of deterioration and learning include Wang et al. [21], Huang et al. [22], Wang et al. [23,24], etc. There is some similarity between the serial-batching problems and group scheduling problems, i.e., the jobs in a batch or group are processed one after another. However, several significant differences can be also found between them, as concluded in Table 1.

In this paper, we further combine the effects of time-dependent deterioration and position-dependent learning to constitute a new model for the job processing time on the serial-batching problems, where the time-dependent setup time is considered simultaneously. To the best of the authors' knowledge, research on serial-batching scheduling with simultaneous considerations of these effects has not been considered so far, but it inevitably happens in many production situations.

The reminder of this paper is organized as follows. The notation and problem description are given in Sect. 2. The solution procedures of minimizing the makespan, the number of tardy jobs, and the maximum lateness are described in Sects. 3–5, respectively. Finally, the conclusion is given in Sect. 6.

2 Notation and problem description

The notation used in this paper is first described as Table 2.

There are n non-preemptive jobs to be processed on a single serial-batching machine. All jobs are first partitioned into multiple serial batches, and then the batches are processed on a single machine. Serial batches require that all the jobs within the same batch are processed one after another in a serial fashion [25], and the completion times of all jobs are equal to that of their batch, which is defined as the completion time of the last job in the batch. The capacity of the serial-batching machine is defined as c, i.e., the maximum number of jobs in a batch is equal to c. A time-dependent setup time precedes the processing of each batch. All jobs are available at time t_0, where $t_0 > 0$. In this paper, we continue our previous model [13] and extend Lee's model [26] in the serial-batching scheduling problem, and the deteriorating and learning effects are both considered in this model. The actual processing time of J_i is indicated as a function of its starting time t such that

$$p_i = \alpha_i t r^a, \quad i = 1, 2, \ldots, n$$

where $t \geq t_0$ is the staring time of processing J_i, $a_i > 0$ is the deterioration rate of J_i, and r is the position of J_i, and $a \leq 0$ is the learning index, respectively.

Table 2 The list of the notation

n	The number of jobs
J_i	Job i, $i = 1, 2, \ldots, n$
p_i	The actual processing time of J_i, $i = 1, 2, \ldots, n$
a_i	The deteriorating rate of J_i, $i = 1, 2, \ldots, n$
a	The learning index of processing time
m	The number of batches
b_k	Batch k, $k = 1, 2, \ldots, m$
n_k	The number of jobs in b_k, $k = 1, 2, \ldots, m$
$S(b_k)$	The starting time of b_k, $k = 1, 2, \ldots, m$
$C(b_k)$	The completion time of b_k, $k = 1, 2, \ldots, m$
θ	The deteriorating rate of the setup time
s_k	The setup time of b_k, $k = 1, 2, \ldots, m$
d	The common due date of all jobs
c	The capacity of the serial-batching machine, i.e., the maximum number of jobs in a batch
π	A schedule of n jobs
$C_i(\pi)$	The completion time of J_i in a given schedule π, $i = 1, 2, \ldots, n$
C_{max}	The makespan of all jobs
$\sum_{i=1}^{n} U_i$	The number of tardy jobs
E_{max}	The maximum earliness of all jobs
$\lceil x \rceil$	The ceiling function of x

Moreover, as in [27], the setup time for b_k is also defined as a simple linear function of its starting time t, that is,

$$s = \theta t$$

where $\theta > 0$ is the deterioration rate of the setup time.

Using the traditional notation, we adopt $C_{max} = \max_{i=1,2,\ldots,n}\{C_i\}$, $E_{max} = \max_{i=1,2,\ldots,n}\{0, d - C_i\}$, and $\sum_{i=1}^{n} U_i$ to represent the makespan, maximum earliness, and total number of tardy jobs, respectively. All jobs have a common due date, where $U_i = 1$ if $C_i > d$ and 0 otherwise. In the remaining sections of the paper, all the problems considered are denoted using the three-field notation schema $\alpha|\beta|\gamma$ introduced by Graham et al. [28].

3 Problem $1|s\text{-}batch,\ p_i = \alpha_i t r^a,\ s = \theta t|C_{max}$

We start with studying the makespan minimization problem $1|s\text{-}batch,\ p_i = a_i t r^a,\ s = \theta t|$ C_{max}. In the following, some properties of the makespan minimization problem are given for any schedule π, and then an optimization algorithm is developed to solve this problem.

Lemma 1 *For any given schedule $\pi = (b_1, b_2, \ldots, b_m)$, if the first batch b_1 starts being processed at time $t_0 > 0$, then the makespan of schedule π is*

$$C_{max}(\pi) = t_0 (1+\theta)^m \prod_{i=1}^{n} \left(1 + \alpha_i i^a\right) \tag{1}$$

Proof The mathematical induction can be used to prove this lemma based on the number of batches. First for $m = 1$, it can be derived that

$$C(b_1) = t_0 + s_1 + \sum_{i=1}^{n_1} p_i = t_0 (1+\theta) \prod_{i=1}^{n_1} \left(1 + \alpha_i i^a\right).$$

Thus, Eq. (1) holds for $m=1$. Suppose for all $2 \le l \le m-1$, Eq. (1) is satisfied. We have

$$C(b_l) = t_0 (1+\theta)^m \prod_{i=1}^{\sum_{f=1}^{l} n_f} \left(1+\alpha_i i^a\right).$$

Then, for the $(l+1)$th batch b_{l+1},

$$\begin{aligned}
C(b_{l+1}) &= C(b_l) + s_{l+1} + \sum_{i=1+\sum_{k=1}^{l} n_k}^{\sum_{f=1}^{l+1} n_f} p_i \\
&= \left[t_0 (1+\theta)^l \prod_{i=1}^{\sum_{f=1}^{l} n_f} \left(1+\alpha_i i^a\right) + t_0 (1+\theta)^l \prod_{i=1}^{\sum_{f=1}^{l} n_f} \left(1+\alpha_i i^a\right) \cdot \theta \right] \\
&\quad \times \prod_{i=1+\sum_{k=1}^{l} n_k}^{\sum_{f=1}^{l+1} n_f} \left(1+\alpha_i i^a\right) \\
&= t_0 (1+\theta)^{l+1} \prod_{i=1}^{\sum_{f=1}^{l} n_f} \left(1+\alpha_i i^a\right) \cdot \prod_{i=1+\sum_{k=1}^{l} n_k}^{\sum_{f=1}^{l+1} n_f} \left(1+\alpha_i i^a\right) \\
&= t_0 (1+\theta)^{l+1} \prod_{i=1}^{\sum_{f=1}^{l+1} n_f} \left(1+\alpha_i i^a\right).
\end{aligned}$$

Hence, Eq. (1) holds for $m = l+1$. since $C_{max}(\pi) = C(b_m)$, the lemma is proved by the induction. □

Lemma 2 *For the problem $1|s\text{-}batch, p_i = \alpha_i t r^a, s = \theta t|C_{max}$, the jobs from the same batch should be sequenced in the non-decreasing order of α_i in an optimal schedule.*

Proof Let π^* and π be an optimal schedule and a job schedule. The difference between the two schedules is the pairwise interchange of these two jobs J_r and $J_{r+1} (r = 1, 2, \ldots, n-1)$, that is, $\pi^* = (W_1, J_r, J_{r+1}, W_2)$, $\pi = (W_1, J_{r+1}, J_r, W_2)$, where $J_r \in b_p$ and $J_{r+1} \in b_p$, $n_p \ge 2$, $p = 1, 2, \ldots, m$. W_1 and W_2 represent two partial sequences, and W_1 or W_2 may be empty. We assume that $a_r > a_{r+1}$.

For π^*, the completion time of b_p is

$$C\left(b_p\left(\pi^*\right)\right) = t_0 (1+\theta)^p \prod_{i=1}^{\sum_{k=1}^{p} n_k} \left(1+\alpha_i i^a\right).$$

For π, the completion time of b_p is

$$C\left(b_p(\pi)\right) = \frac{\left(1+\alpha_{r+1} r^a\right)\left[1+\alpha_r (r+1)^a\right]}{\left(1+\alpha_r r^a\right)\left[1+\alpha_{r+1} (r+1)^a\right]} \cdot t_0 (1+\theta)^p \prod_{i=1}^{\sum_{k=1}^{p} n_k} \left(1+\alpha_i i^a\right).$$

Then,

$$
\begin{aligned}
&C\left(b_p\left(\pi^*\right)\right)-C\left(b_p(\pi)\right)\\
&=\left[1-\frac{\left(1+\alpha_{r+1}r^a\right)\left[1+\alpha_r(r+1)^a\right]}{\left(1+\alpha_r r^a\right)\left[1+\alpha_{r+1}(r+1)^a\right]}\right]\cdot t_0(1+\theta)^p\prod_{i=1}^{\sum_{k=1}^{p}n_k}\left(1+\alpha_i i^a\right)\\
&=\frac{\left(\alpha_r-\alpha_{r+1}\right)\left[r^a-(r+1)^a\right]}{\left(1+\alpha_r r^a\right)\left[1+\alpha_{r+1}(r+1)^a\right]}\cdot t_0(1+\theta)^p\prod_{i=1}^{\sum_{k=1}^{p}n_k}\left(1+\alpha_i i^a\right).
\end{aligned}
$$

Since $a\leq 0$, we have $r^a\geq(r+1)^a$. It can be derived that

$$
\frac{\left(\alpha_r-\alpha_{r+1}\right)\left[r^a-(r+1)^a\right]}{\left(1+\alpha_r r^a\right)\left[1+\alpha_{r+1}(r+1)^a\right]}\geq 0.
$$

Thus,

$$
C\left(b_p\left(\pi^*\right)\right)\geq C\left(b_p(\pi)\right),
$$

which conflicts with the optimal schedule. Hence, $a_r\leq a_{r+1}$. This proves the lemma. □

Lemma 3 *For the problem* $1|s\text{-}batch, p_i=\alpha_i t r^a, s=\theta t|C_{max}$, *considering two jobs* $J_u\in b_p$ *and* $J_v\in b_{p+1}$ *from two consecutive batches, it should be* $a_u\leq a_v$ *in an optimal schedule.*

Proof Let π^* and π be an optimal schedule and a job schedule. The difference between the two schedules is the pairwise interchange of these two jobs J_u and J_v (where J_u and J_v are in the uth and vth positions, and $u<v$), that is, $\pi^*=(W_1,b_p,b_{p+1},W_2)$, $\pi=(W_1,(b_p/\{J_u\})\cup\{J_v\},(b_{p+1}/\{J_v\})\cup\{J_u\},W_2)$. W_1 and W_2 represent two partial sequences, and W_1 or W_2 may be empty. We assume that $a_u>a_v$.

For π^*, the completion time of b_{p+1} is

$$
C\left(b_{p+1}\left(\pi^*\right)\right)=t_0(1+\theta)^{p+1}\prod_{i=1}^{\sum_{k=1}^{p+1}n_k}\left(1+\alpha_i i^a\right).
$$

For π, the completion time of b_{p+1} is

$$
C\left(b_{p+1}(\pi)\right)=t_0(1+\theta)^{p+1}\prod_{i=1}^{\sum_{k=1}^{p+1}n_k}\left(1+\alpha_i i^a\right)\cdot\frac{1+\alpha_v u^a}{1+\alpha_u u^a}\cdot\frac{1+\alpha_u v^a}{1+\alpha_v v^a}.
$$

Then,

$$
\begin{aligned}
&C\left(b_{p+1}\left(\pi^*\right)\right)-C\left(b_{p+1}(\pi)\right)\\
&=\left[1-\frac{1+\alpha_v u^a}{1+\alpha_u u^a}\cdot\frac{1+\alpha_u v^a}{1+\alpha_v v^a}\right]\cdot t_0(1+\theta)^{p+1}\prod_{i=1}^{\sum_{k=1}^{p+1}n_k}\left(1+\alpha_i i^a\right)\\
&=\frac{\left(u^a-v^a\right)\left(a_u-a_v\right)}{\left(1+\alpha_u u^a\right)\left(1+\alpha_v v^a\right)}\cdot t_0(1+\theta)^{p+1}\prod_{i=1}^{\sum_{k=1}^{p+1}n_k}\left(1+\alpha_i i^a\right)
\end{aligned}
$$

Since $a_u>a_v$ and $u^a\geq v^a$, we have $(u^a-v^a)(a_u-a_v)\geq 0$. Thus, it can be derived that $C(b_{p+1}(\pi^*))\geq C(b_{p+1}(\pi))$, which conflicts with the optimal solution. Hence, it should be $a_u\leq a_v$ in an optimal schedule. This completes the proof. □

Based on Lemmas 2 and 3, we have the following corollary.

Corollary 1 *For the problem* $1|s\text{-}batch,\ p_i = \alpha_i tr^a, s = \theta t|C_{max}$*, all jobs should be sequenced in the non-decreasing order of* a_i*.*

Lemma 4 *For the problem* $1|s\text{-}batch,\ p_i = \alpha_i tr^a, s = \theta t|C_{max}$*, there should be* $\lceil \frac{n}{c} \rceil$ *batches in the optimal schedule.*

Proof Based on Lemma 1 and Corollary 1, the makespan of the optimal schedule is dependent on $(1+\theta)^m$ after all jobs are sequenced in the non-decreasing order of a_i. Since $(1+\theta)^m \geq (1+\theta)^{\lceil \frac{n}{c} \rceil}$, it is easily derived that there should be $\lceil \frac{n}{c} \rceil$ batches in the optimal schedule. □

Corollary 2 *For the problem* $1|s\text{-}batch,\ p_i = \alpha_i tr^a, s = \theta t|C_{max}$*, all the batches, except possibly the highest indexed one, are full in an optimal schedule.*

Based on the above lemmas and corollaries, we design the following Algorithm 1 to solve the problem $1|s\text{-}batch,\ p_i = \alpha_i tr^a, s = \theta t|C_{max}$.

Algorithm 1

Step 1. All jobs are indexed in the non-decreasing order of a_i such that $a_1 \leq a_2 \leq \cdots \leq a_n$, and a job list is obtained
Step 2 If there are more than c jobs in the job list, then place the first c jobs in a batch and iterate. Otherwise, place the remaining jobs in a batch
Step 3 The batches are scheduled in their generation order at time t_0

Theorem 1 *For the problem* $1|s\text{-}batch,\ p_i = \alpha_i tr^a, s = \theta t|C_{max}$*, an optimal schedule can be obtained by Algorithm 1 in* $O(n \log n)$ *time, and the optimal makespan is*

$$C^*_{max} = t_0(1+\theta)^{\lceil \frac{n}{c} \rceil} \prod_{i=1}^{n} \left(1 + \alpha_i i^a\right) \tag{2}$$

Proof Based on Lemmas 1–4 and Corollaries 1 and 2, an optimal solution can be generated by Algorithm 1. We can also obtain the result of the optimal solution as Eq. (2). The time complexity of step 1 is $O(n \log n)$, and the time complexity of step 2 is at most $O(n)$. Then, the total time complexity of Algorithm 1 is $O(n \log n)$. The proof is completed. ■

4 Problem $1|s\text{-}batch,\ p_i = \alpha_i tr^a, s = \theta t| \sum_{i=1}^{n} U_i$

In the following section, some properties are first given for the problem of minimizing the number of tardy jobs, and then an $O(n \log n)$ time algorithm is proposed to solve this problem. The job sets of which the completion times are no more than and more than the common due date d are denoted as O (i.e., ordinary jobs) and L (i.e., late jobs), respectively.

Lemma 5 *For the problem* $1|s\text{-}batch,\ p_i = \alpha_i tr^a, s = \theta t| \sum_{i=1}^{n} U_i$*, an optimal schedule satisfies the following properties:*

(1) *All jobs should be sequenced in the non-decreasing order of* a_i *in* O*;*
(2) *All batches are full except possibly the highest indexed batch in* O*;*

(3) *If there exists a batch* b_k *in* L *satisfying that* $S(b_k) < d$ *and* $C(b_k) > d$ *,then it should be* $C(b_{k-1})(1+\theta)(1+\alpha_i i^a) > d$ *for an arbitrary job* $J_i \in L$.

(4) *The deterioration rate of an arbitrary job in* O *is no more than that of other jobs in* L.

Proof (1) The proof is similar with that of Lemmas 2 and 3, and we omit it.

(2) Based on Lemma 4, this property can be proved.

(3) We assume that there exists a batch b_k in L satisfying that $S(b_k) < d$ and $C(b_k) > d$, and the condition that$C(b_{k-1})(1+\theta)(1+\alpha_i i^a) \le d$ is satisfied for a job $J_i \in L$. After J_i is placed into a single batch, the solution can be improved, which contradicts with the optimal schedule. Thus, it can be derived that $C(b_{k-1})(1+\theta)(1+\alpha_i i^a) > d$ for an arbitrary job $J_i \in L$.

(4) Let π^* and π be an optimal schedule and a job schedule. The difference between the two schedules is the pairwise interchange of these two jobs J_u and J_v (where J_u and J_v are in the uth and vth positions, and $u < v$), that is, $\pi^* = (W_1, b_p, W_2, b_q, W_3)$, $\pi = (W_1, (b_p/\{J_u\}) \cup \{J_v\}, W_2, (b_q/\{J_v\}) \cup \{J_u\}, W_3)$. W_1, W_2 and W_3 represent three partial sequences, where W_1, W_2, or W_3 may be empty, $b_p \subseteq O$, and $b_q \subseteq L$. Here we assume that $a_u > a_v$.

The completion time of b_p in π^* is

$$C\left(b_p\left(\pi^*\right)\right) = t_0\left(1+\theta\right)^p \prod_{i=1}^{\sum_{k=1}^{p} n_k} \left(1+\alpha_i i^a\right).$$

The completion time of b_p in π is

$$C\left(b_p\left(\pi\right)\right) = t_0\left(1+\theta\right)^p \prod_{i=1}^{\sum_{k=1}^{p} n_k} \left(1+\alpha_i i^a\right) \cdot \frac{1+\alpha_v u^a}{1+\alpha_u u^a}.$$

Then,

$$\begin{aligned} C\left(b_p\left(\pi^*\right)\right) - C\left(b_p(\pi)\right) &= t_0\left(1+\theta\right)^p \prod_{i=1}^{\sum_{k=1}^{p} n_k} \left(1+\alpha_i i^a\right) \cdot \left(1 - \frac{1+\alpha_v u^a}{1+\alpha_u u^a}\right) \\ &= t_0\left(1+\theta\right)^p \prod_{i=1}^{\sum_{k=1}^{p} n_k} \left(1+\alpha_i i^a\right) \cdot \frac{u^a\left(\alpha_u - \alpha_v\right)}{1+\alpha_u u^a}. \end{aligned}$$

Since $a_u > a_v$, it can be derived that $C(b_p(\pi^*)) \ge C(b_p(\pi))$, which conflicts with the optimal solution. Hence, it should be $a_u \le a_v$ in an optimal schedule. This completes the proof. □

Based on the above Lemma 5, the following Algorithm 2 is developed to solve the problem $1|s\text{-}batch, p_i = \alpha_i t r^a, s = \theta t| \sum_{i=1}^{n} U_i$.

Theorem 2 *For the problem* $1|s\text{-}batch, p_i = \alpha_i t r^a, s = \theta t| \sum_{i=1}^{n} U_i$, *an optimal schedule can be obtained by Algorithm 2 in* $O(n \log n)$ *time.*

Proof Based on Lemma 5, Algorithm 2 can generate an optimal solution. The time complexity of step 1 is $O(n \log n)$, the total time complexity of steps 2, 3, 4, 5, 6, and 7 is $O(n)$. Thus, the time complexity of Algorithm 2 is at most $O(n \log n)$. □

Algorithm 2

Step 1. All jobs are indexed in the non-decreasing order of a_i such that $a_1 \leq a_2 \leq \cdots \leq a_n$, and a job list is obtained
Step 2. Initialize the first batch, and set $k = 1$, $i = 1$, $b_k = \{J_i\}$, $n_k = 1$, $C(b_k) = t_0(1+\theta)(1+\alpha_i i^a)$
Step 3. If $i < n$, then update $i = i + 1$ and go to step 4. Otherwise, go to step 7
Step 4. If $n_k < c$, then go to step 5. Otherwise, go to step 6
Step 5. If $C(b_k)(1+\alpha_i i^a) \leq d$, then update $b_k = b_k \cup \{J_i\}$, $n_k = n_k + 1$, and $C(b_k) = C(b_k)(1+\alpha_i i^a)$. Otherwise, go to step 7
Step 6. If $C(b_k)(1+\theta)(1+\alpha_i i^a) \leq d$, then update $k = k + 1$, $b_k = \{J_i\}$, $n_k = 1$, and $C(b_k) = C(b_k)(1+\theta)(1+\alpha_i i^a)$. Otherwise, go to step 7
Step 7. Batch the remaining jobs arbitrarily and output the schedule of batches as their generation sequence

5 Problem $1|s\text{-}batch,\, p_i = \alpha_i t r^a,\, s = \theta t|\, E_{max}$

In this section, the problem of minimizing the maximum earliness of all jobs is studied. As commonly assumed in the previous work involving earliness [29], all jobs are restricted to be completed no earlier than the common due date d, otherwise each job can be trivially scheduled sufficiently late to avoid earliness cost. Thus, it should be $d \geq t_0 + \sum_{k=1}^{m} s_k + \sum_{i=1}^{n} p_i$. The earliness of job J_i is given by $E_i = \{0, d - C_i\}$, and the maximum earliness is defined as $E_{max} = \max_{i=1,2,\ldots,n} E_i$. Let π be a feasible schedule. Some properties are first developed as follows.

Lemma 6 *For the problem* $1|s\text{-}batch, p_i = \alpha_i t r^a, s = \theta t|E_{max}$, *there is an optimal schedule with the first batch* b_1 *starting at time* t_0 *such that* $t_0(1+\theta)^m \prod_{i=1}^{n}(1+\alpha_i i^a) = d$, *and there are no idle times between consecutive batches or consecutive jobs in the same batch.*

Proof All jobs need to be scheduled as late as possible and also satisfy the constraint that their completion times should be no earlier than the common due date. Hence, the completion time of the last batch should be just equal to the common due date. Then, we have $t_0(1+\theta)^m \prod_{i=1}^{n}(1+\alpha_i i^a) = d$ based on Lemma 1. In addition, if there is any idle time between consecutive batches or consecutive jobs in the same batch, then the maximum earliness will be increased as the starting time t_0 gets smaller. Thus, there are no idle times between them.

Lemma 7 *For the problem* $1|s\text{-}batch, p_i = \alpha_i t r^a, s = \theta t|E_{max}$, *if* $n \leq c$, *then there should be only one batch in the optimal schedule, and the optimal maximum earliness is* $E_{max}^* = 0$.

Proof Since the completion time of a job is equal to the completion time of its batch, the maximum earliness will be decreased as the processing time of the first batch becomes longer. If $n \leq c$, then it can be derived that all jobs should be arranged in the first batch. Thus, there is only one batch, i.e., $n_1 = n$. Based on Lemma 6, the first batch b_1 should start at time t_0 such that $t_0(1+\theta)\prod_{i=1}^{n}(1+\alpha_i i^a) = d$. Hence, the completion time of all jobs in b_1 is equal to d, and $E_{max}^* = 0$. □

Lemma 8 *For the problem* $1|s\text{-}batch, p_i = \alpha_i t r^a, s = \theta t|E_{max}$, *if* $n > c$, *then an optimal schedule satisfies the following properties:*

(1) $m = \lceil \frac{n}{c} \rceil$.
(2) *All jobs should be sequenced in the non-decreasing order of* a_i *starting from* b_2.
(3) $n_1 = c$.
(4) *The deterioration rate of an arbitrary job in* b_1 *is no smaller than that of other batches.*

Proof Based on Lemma 6, it can be derived that

$$t_0 = \frac{d}{(1+\theta)\prod_{i=1}^{n_1}(1+\alpha_i i^a)\cdot(1+\theta)^{m-1}\prod_{i=n_1+1}^{n}(1+\alpha_i i^a)}.$$

Then,

$$E_{max} = d - t_0(1+\theta)\prod_{i=1}^{n_1}\left(1+\alpha_i i^a\right) = d - \frac{d}{(1+\theta)^{m-1}\prod_{i=n_1+1}^{n}(1+\alpha_i i^a)}.$$

Since d is a constant, minimizing E_{max} is equivalent to minimizing the result of $(1+\theta)^{m-1}\prod_{i=n_1+1}^{n}(1+\alpha_i i^a)$, and we have $(1+\theta)^{m-1} \geq (1+\theta)^{\lceil \frac{n}{c}\rceil-1}$. In order to minimize the result of $(1+\theta)^{m-1}\prod_{i=n_1+1}^{n}(1+\alpha_i i^a)$, the number of batches should be equal to the least batch number $\lceil \frac{n}{c}\rceil$, the number of the first batch should be c, and the deterioration rate of an arbitrary job in b_1 is no smaller than that of other batches. Based on Corollary 1, all jobs should be sequenced in the non-decreasing order of a_i starting from b_2. The proof is completed. □

Based on Lemmas 7 and 8, we design the following Algorithm 3 to solve this problem.

Algorithm 3

Step 1. If $n \leq c$, then arrange all jobs in a batch and output the schedule. Otherwise, go to step 2
Step 2. Index all jobs in the non-decreasing order of a_i such that $a_1 \leq a_2 \leq \cdots \leq a_n$, and a job list is obtained
Step 3. Arrange the last n jobs into a batch from the job list, and update the job list
Step 4. If there are more than c jobs in the job list, then place the first c jobs in a batch and iterate. Otherwise, place the remaining jobs in a batch
Step 5. The batches are scheduled in their generation order, and these batches are processed at time $t_0 = \frac{d}{(1+\theta)^{\lceil \frac{n}{c}\rceil}\prod_{i=n-c+1}^{n}(1+\alpha_i(i-n+c)^a)\cdot\prod_{i=1}^{n-c}(1+\alpha_i(i+c)^a)}$

Theorem 3 *For the problem* $1|s\text{-}batch, p_i = \alpha_i t r^a, s = \theta t|E_{max}$, *an optimal schedule can be obtained by Algorithm 3 in* $O(n\log n)$ *time. If all jobs are indexed in the non-decreasing order of* a_i *such that* $a_1 \leq a_2 \leq \cdots \leq a_n$, *then the optimal maximum earliness is*

$$E^*_{max} = d - \frac{d}{(1+\theta)^{\lceil \frac{n}{c}\rceil-1}\prod_{i=1}^{n-c}\left(1+\alpha_i\,(i+c)^a\right)}.$$

Proof Based on Lemmas 7 and 8, Algorithm 3 can generate an optimal schedule for this problem, and the optimal maximum earliness is

$$\begin{aligned}
E^*_{max} &= d - C\,(b_1)\\
&= d - t_0\,(1+\theta)\prod_{i=n-c+1}^{n}\left(1+\alpha_i\,(i-n+c)^a\right)\\
&= d - t_0\,(1+\theta)\prod_{i=n-c+1}^{n}\left(1+\alpha_i\,(i-n+c)^a\right)\\
&\quad\times\frac{d}{(1+\theta)^{\lceil \frac{n}{c}\rceil}\prod_{i=n-c+1}^{n}\left(1+\alpha_i\,(i-n+c)^a\right)\cdot\prod_{i=1}^{n-c}\left(1+\alpha_i\,(i+c)^a\right)}\\
&= d - \frac{d}{(1+\theta)^{\lceil \frac{n}{c}\rceil-1}\prod_{i=1}^{n-c}\left(1+\alpha_i\,(i+c)^a\right)}.
\end{aligned}$$

The time complexity of step 1 is $O(1)$, the time complexity of step 2 is $O(n \log n)$, and the total time complexity of steps 3, 4, and 5 is no more than $O(n)$. Thus, and the total time complexity of Algorithm 3 is $O(n \log n)$. □

6 Conclusions

In this paper, we study a single serial-batching machine scheduling problem with deteriorating jobs and learning effect, where the deterioration of set-up time is also considered. Although the concept of deteriorating jobs and the learning effect have been extensively studied, they still have not been considered in the serial-batching scheduling problem simultaneously. In this case, we first derive some properties for the problems of minimizing the makespan, the number of tardy jobs, and the maximum earliness, and then three optimization algorithms are developed to solve these problems, respectively. In future research, we can investigate more general serial-batching scheduling models with deteriorating jobs and learning effect, consider more objective functions, and extend these problems to multiple machines.

Acknowledgments This work is supported by the National Natural Science Foundation of China (Nos. 71231004, 71171071, 71131002, 71201042), and the Fundamental Research Funds for the Central Universities (Nos. JZ2015HGBZ0116, JZ2015HGBZ0117, JZ2015HGQC0187). Panos M. Pardalos is partially supported by the project of "Distinguished International Professor by the Chinese Ministry of Education" (MS2014HFGY026).

References

1. Biskup, D.: Single-machine scheduling with learning considerations. Eur. J. Oper. Res. **115**(1), 173–178 (1999)
2. Cheng, T.C.E., Wang, G.: Single machine scheduling with learning effect considerations. Ann. Oper. Res. **98**(1–4), 273–290 (2000)
3. Mosheiov, G., Sidney, J.B.: Scheduling with general job-dependent learning curves. Eur. J. Oper. Res. **147**(3), 665–670 (2003)
4. Gupta, J.N.D., Gupta, S.K.: Single facility scheduling with nonlinear processing times. Comput. Ind. Eng. **14**(4), 387–393 (1988)
5. Browne, S., Yechiali, U.: Scheduling deteriorating jobs on a single processor. Oper. Res. **38**(3), 495–498 (1990)
6. Cheng, T.C.E., Ding, Q., Lin, B.M.T.: A concise survey of scheduling with time-dependent processing times. Eur. J. Oper. Res. **152**(1), 1–13 (2004)
7. Gawiejnowicz, S.: Time-Dependent Scheduling, Monographs in Theoretical Computer Science, an EATCS Series. Springer, Berlin (2008)
8. Jafari, A., Moslehi, G.: Scheduling linear deteriorating jobs to minimize the number of tardy jobs. J. Glob. Optim. **54**(2), 389–404 (2012)
9. Qi, X.L., Zhou, S.G., Yuan, J.J.: Single machine parallel-batch scheduling with deteriorating jobs. Theor. Comput. Sci. **410**(8–10), 830–836 (2009)
10. Yang, D.-L., Kuo, W.-H.: A single-machine scheduling problem with learning effects in intermittent batch production. Comput. Ind. Eng. **57**(3), 762–765 (2009)
11. Li, S.S., Ng, C.T., Cheng, T.C.E., Yuan, J.J.: Parallel-batch scheduling of deteriorating jobs with release dates to minimize the makespan. Eur. J. Oper. Res. **210**(3), 482–488 (2011)
12. Miao, C.X., Zhang, Y.Z., Wu, C.L.: Scheduling of deteriorating jobs with release dates to minimize the maximum lateness. Theor. Comput. Sci. **462**(30), 80–87 (2012)
13. Pei, J., Pardalos, P.M., Liu, X., Fan, W., Yang, S.: Serial batching scheduling of deteriorating jobs in a two-stage supply chain to minimize the makespan. Eur. J. Oper. Res. **244**(1), 13–25 (2015)
14. Cheng, T.C.E., Lee, W.-C., Wu, C.-C.: Scheduling problems with deteriorating jobs and learning effects including proportional setup times. Comput. Ind. Eng. **58**(2), 326–331 (2010)

15. Wang, J.-B., Jiang, Y., Wang, G.: Single-machine scheduling with past-sequence-dependent setup times and effects of deterioration and learning. Int. J. Adv. Manuf. Technol. **41**(11–12), 1221–1226 (2009)
16. Wu, C.-C., Lee, W.-C.: Single-machine group-scheduling problems with deteriorating setup times and job-processing times. Int. J. Prod. Econ. **115**(1), 128–133 (2008)
17. Pei, J., Liu, X., Fan, W., Pardalos, P.M., Migdalas, A., Yang, S.: Scheduling jobs on a single serial-batching machine with dynamic job arrivals and multiple job types. Ann. Math. Artif. Intell. doi:10.1007/s10472-015-9449-7 (2015)
18. Pei, J., Liu, X., Pardalos, P.M., Fan, W., Yang, S., Wang, L.: Application of an effective modified gravitational search algorithm for the coordinated scheduling problem in a two-stage supply chain. Int. J. Adv. Manuf. Technol. **70**(1–4), 335–348 (2014)
19. Yang, S.J.: Group scheduling problems with simultaneous considerations of learning and deterioration effects on a single-machine. Appl. Math. Model. **35**(8), 4008–4016 (2011)
20. Bai, J., Li, Z.R., Huang, X.: Single-machine group scheduling with general deterioration and learning effects. Appl. Math. Model. **36**(3), 1267–1274 (2012)
21. Wang, J.B., Gao, W.J., Wang, L.Y., Wang, D.: Single machine group scheduling with general linear deterioration to minimize the makespan. Int. J. Adv. Manuf. Technol. **43**, 146–150 (2009)
22. Huang, X., Wang, M.Z., Wang, J.B.: Single-machine group scheduling with both learning effects and deteriorating jobs. Comput. Ind. Eng. **60**, 750–754 (2011)
23. Wang, J.B., Huang, X., Wu, Y.B., Ji, P.: Group scheduling with independent setup times, ready times, and deteriorating job processing times. Int. J. Adv. Manuf. Technol. **60**, 643–649 (2012)
24. Wang, D., Huo, Y.Z., Ji, P.: Single-machine group scheduling with deteriorating jobs and allotted resource. Optim. Lett. **8**, 591–605 (2014)
25. Xuan, H., Tang, L.X.: Scheduling a hybrid flowshop with batch production at the last stage. Comput. Oper. Res. **34**(9), 2718–2733 (2007)
26. Lee, W.C.: A note on deteriorating jobs and learning in single-machine scheduling problems. Int. J. Bus. Econ. **3**(1), 83–89 (2004)
27. Cheng, T.C.E., Hsu, C.-J., Huang, Y.-C., Lee, W.-C.: Single-machine scheduling with deteriorating jobs and setup times to minimize the maximum tardiness. Comput. Oper. Res. **38**(12), 1760–1765 (2011)
28. Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann. Discret. Math. **5**, 287–326 (1979)
29. Yin, Y., Cheng, S.R., Wu, C.C.: Scheduling problems with two agents and a linear non-increasing deterioration to minimize earliness penalties. Inf. Sci. **189**, 282–292 (2012)