Optim Lett (2017) 11:1257–1271
DOI 10.1007/s11590-016-1074-9

ORIGINAL PAPER

Single-machine serial-batching scheduling with a machine availability constraint, position-dependent processing time, and time-dependent set-up time

Jun Pei[1,2] · **Xinbao Liu**[1,3] · **Panos M. Pardalos**[2] · **Kai Li**[1,3] · **Wenjuan Fan**[1,3] · **Athanasios Migdalas**[4,5]

Received: 23 January 2016 / Accepted: 25 August 2016 / Published online: 6 September 2016

Abstract This article considers the single-machine serial-batching scheduling problem with a machine availability constraint, position-dependent processing time, and time-dependent set-up time. The objective of this problem is to make the decision of batching jobs and sequencing batches to minimize the makespan. To solve the problem, three cases of machine non-availability periods are considered, and the structural properties of the optimal solution are derived for each case. Based on these structural properties, an optimization algorithm is developed and an example is proposed to illustrate this algorithm.

Keywords Scheduling · Availability constraint · Serial-batching · Single-machine · Position-dependent processing time

✉ Jun Pei
feiyijun.ufl@gmail.com

Xinbao Liu
lxb@hfut.edu.cn

Wenjuan Fan
fanwenjuan@hfut.edu.cn

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, USA

3 Key Laboratory of Process Optimization and Intelligent Decision-making of 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, Thessaloniki, Greece

1 Introduction

In classic deterministic scheduling models, the job processing times are treated as certain given constants, but this is not appropriate for all actual situations. There are many real-life production situations where the job processing times may vary over time due to the learning effect. Biskup [1] considered the learning effect in the scheduling problems for the first time, and he pointed out that the learning effects have been observed in numerous practical situations in different branches of industry. However, many studies on the scheduling problem with the learning effect assumed that the machines are available all the time. This assumption may not be true in real production settings. Machine breakdown and preventive maintenance may occur during the scheduling period. In this paper we investigate the scheduling problem on a single serial-batching machine with the learning effect and an availability constraint for the deterministic case. To the best of our knowledge, this is the first attempt to investigate this type of problem.

Wright [2] first found the impact of learning on productivity in the aircraft industry. Then, Biskup [1] and Chen and Wang [3] are among the pioneers that combined the learning effect and scheduling fields. Afterwards, more researches have been devoted to investigating the learning effects in many scheduling situations. Recently, Wang and Wang [4] studied a new scheduling model with sum-of-logarithm-processing-times based and position based learning effects. Based on the proposed model, they show that some single-machine scheduling problems are still polynomially solvable. Lee [5] studied a scheduling model where the learning effect, deteriorating jobs, and the setup times are considered simultaneously. Some optimal schedules are proposed for some single-machine scheduling problems. Cheng et al. [6] introduced a scheduling model with a position-weighted learning effect. They develop some optimal solutions to minimize the makespan and the total completion time, and an optimal solution is also provided to minimize the total tardiness under an agreeable situation. Wang and Wang [7] considered flowshop scheduling problems with a general exponential learning effect, and several heuristic algorithms are developed for five objective functions: the makespan, the total completion time, the total weighted completion time, the total weighted discounted completion time, and the sum of the quadratic job completion times. The worst-case bound of each heuristic algorithm is analyzed. Luo and Zhang [8] investigated a simple proof technique to analyze the scheduling models with the learning effects. Based on the famous Lagrange mean value theorem, a simpler technique is proposed to simplify the proofs. Based on a common flow allowance, Li et al. [9] studied a single-machine due-window assignment scheduling problem with learning effect and controllable processing times, and the objective is to minimize a linear combination of earliness, tardiness, window location, window size, makespan, and resource consumption. A polynomial-time algorithm is developed to solve them, respectively.

The majority of the scheduling literature carries a common assumption that machines are available simultaneously. However, the machine may not always be available in real industrial setting, and it is mainly due to machine breakdowns or preventive maintenance during the scheduling period. Many researchers have focused on this type of the scheduling problems. Based on various performance measures and various machine environments, Lee [10] presented an extensive study of the single and parallel machine scheduling problems with an availability constraint. Lee [11] investigated the model of the two-machine flowshop problem with the availability constraint on one machine and both machines, respectively. Dynamic programming algorithms are developed to solve the problems optimally and heuristic methods are proposed with an error bound analysis. Zhong et al. [12] studied an order acceptance and scheduling model with machine availability constraints, where the machine is available to process orders only within a number of discontinuous time intervals. The objective is to minimize the makespan of all accepted orders plus the total penalty of all rejected/outsourced orders, and the approximability of the model and some of its important special cases are studied. Kacem et al. [13] considered the maximization of the weighted number of early jobs on a single machine with non-availability constraints. Both the resumable and the non-resumable cases are investigated, and some polynomial time approximation schemes are developed to solve them. Mor and Mosheiov [14] studied single-machine scheduling problems with an unavailability constraint and position-dependent processing times. The objectives are to minimize makespan, minimum total completion time, and minimum number of tardy jobs. Heuristics are developed to solve them, respectively, and lower bounds, worst case analysis, and asymptotic optimality are also discussed. Wu and Lee [15] considered a single-machine scheduling problem with learning effect and an availability constraint. They showed that the shortest processing time rule provides the optimal schedules for the makespan and the total completion time minimization problems when jobs are assumed to be resumable.

The batch production is an important class of scheduling problems, and many researchers have investigated the batching scheduling problems, i.e., parallel-batching and serial-batching problems. The parallel-batching scheduling problems with deterioration and/or learning effects have been studied by several researchers in the last decade, including Yang and Kuo [16] and Li et al. [17]. The problem studied in this paper is similar to the group scheduling problem with the effect of learning which has been studied extensively, including Yang [21], Bai et al. [22], and Huang et al. [23]. There are some similarities and differences between group and serial-batching scheduling problems, and we have analyzed in depth in [24]. We also have investigated some serial-batching scheduling problems with deterioration and learning effects. In [18], 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. Two scheduling models with and without buffers are considered, and we developed the optimization algorithm and heuristic algorithm to solve them. In [19], a scheduling model with the features of deteriorating jobs, serial batches, multiple job types, and setup times are considered simultaneously. In [20], the serial-batching scheduling problems with the effects of deterioration and learning are considered simultaneous. This paper differs from our previous research in the following two aspects. Firstly, unlike our previous papers that assume machine availability at all times, in this paper deterministic availability constraint is imposed on the machine according to the practical production situation at the factory. Secondly, we investigate the different position-dependent learning effect as function of the actual job processing time.

The reminder of this paper is organized as follows. The problem description is given in Sect. 2. In Sect. 3, the structure properties of three cases for the makespan minimization problem are analyzed and our solution procedure is presented. Finally, the conclusion is given in Sect. 4.

2 Notations and problem statement

In this section, a new scheduling model is investigated by extending Wang and Xia's idea [25] to the context of serial-batching scheduling with an availability constraint. The model is described as follows. There are given n independent and non-preemptive jobs $J_1, J_2, \ldots, J_n$ to be processed on a single serial-batching machine. All jobs are first partitioned into multiple serial batches, and then these different 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 [26], and the setup operation is needed before processing each batch. The general model of the job processing time is considered from Wang and Xia [25], and it is based on a position-dependent learning effect. That is, when J_i is scheduled in the position r, the actual processing time of job J_i is

$$p_{ir} = p_i\,(a - br) \tag{1}$$

where p_i is the normal processing time, a and b denote a constant number and a learning ratio. Here $a, b > 0$, it is assumed that $a - bn > 0$, and the values of these parameters are determined based on the machine's status and the practical production situations. All jobs are available for processing at time t_0.

The setup operation is used for adjustment of related assistive equipment, and longer setup time might be necessary as the assistive equipment's condition worsens. As in [27], the setup time s for a batch is also defined as a simple linear function of its starting time t, and it is defined as

$$s = \theta t \tag{2}$$

where θ is the deteriorating rate of the setup time.

Let c denote the capacity of the serial-batching machine, i.e., the maximum number of jobs in a batch. Also, we denote the number of batches and the number of jobs in the kth batch b_k as m and n_k, i.e., $n_k \le c$, $k = 1, 2, \ldots, m$. Here we also investigate the situation that the machine is not available from the period d_1 to d_2 [28], where $0 \le d_1 \le d_2$. Thus, resumable jobs are considered in the present paper. If a job is interrupted by the start time of a non-availability period, and it does not need to be restarted and can continue to be processed after the machine becomes available again, then this job is said to be resumable [28,29]. The objective of this paper is to find an optimal schedule to minimize the makespan. For a given schedule $\pi = \{J_1, J_2, \ldots, J_n\}$, let $C_i(\pi)$ and $C(b_k)$ represent the completion times of the job J_i and the batch b_k, respectively. Also, let $C_{max} = max\{C_i | i = 1, 2, \ldots, n\}$ represent the makespan based on the traditional notation and m represent the number of batches in a schedule. The problem under consideration is denoted as $1|s-batch, r-a, p_{ir}=p_i(a-br), s=\theta t|C_{max}$ using the three-field notation schema α $|\beta|\gamma$ introduced by Graham et al. [30].

3 The problem $1|s-batch, r-a, p_{ir}=p_i(a-br), s=\theta t|C_{max}$

In this section, we focus on the makespan minimization problem. Depending on the relation among the starting time of the setup, the completion time of the batch, and the starting time for the machine's non-availability period, these situations are divided into the following three cases.

Case 1. The starting time for the non-availability period is bigger than the starting time of one batch and smaller than the completion time of one batch. Or, the starting time for the non-availability period is bigger than the starting time of one batch's setup and smaller than the completion time of one batch's setup, and this batch is not the first batch.

Case 2. The starting time for the non-availability period is bigger than the starting time of the first batch's setup and smaller than the completion time of the first batch's setup.

Case 3. The starting time for the non-availability period is bigger than the completion time of all batches.
These three cases are shown in Fig. 1.

Then, we discuss on these three cases, respectively.

(1) Case 1

Lemma 1 *For Case 1, given any schedule* $\pi=(b_1, b_2, \ldots, b_m)$*, with the first batch* b_1*starting at time* $t_0>0$*, if the starting time for the non-availability period is bigger than the starting time of* $b_f(0<f\le m)$*and smaller than the completion time of* b_f*, then the makespan of schedule* π *is*

$$C_{max}(\pi)=t_0\left(1+\theta^m\right)+\sum_{k=1}^{m}\sum_{i=1+\sum_{j=1}^{k-1}n_j}^{\sum_{j=1}^{k}n_j}p_i(a-bi)(1+\theta)^{m-k}$$
$$+(d_2-d_1)(1+\theta)^{m-f+1} \tag{3}$$

Proof For the batch index $l=1$, there is

$$C(b_1)=t_0(1+\theta)+\sum_{i=1}^{n_1}p_i(a-bi).$$

For all $2\le l<f$, if Eq. (3) holds, then

$$C(b_l)=t_0(1+\theta)^l+\sum_{k=1}^{l}\sum_{i=1+\sum_{j=1}^{k-1}n_j}^{\sum_{j=1}^{k}n_j}p_i(a-bi)(1+\theta)^{l-k},$$

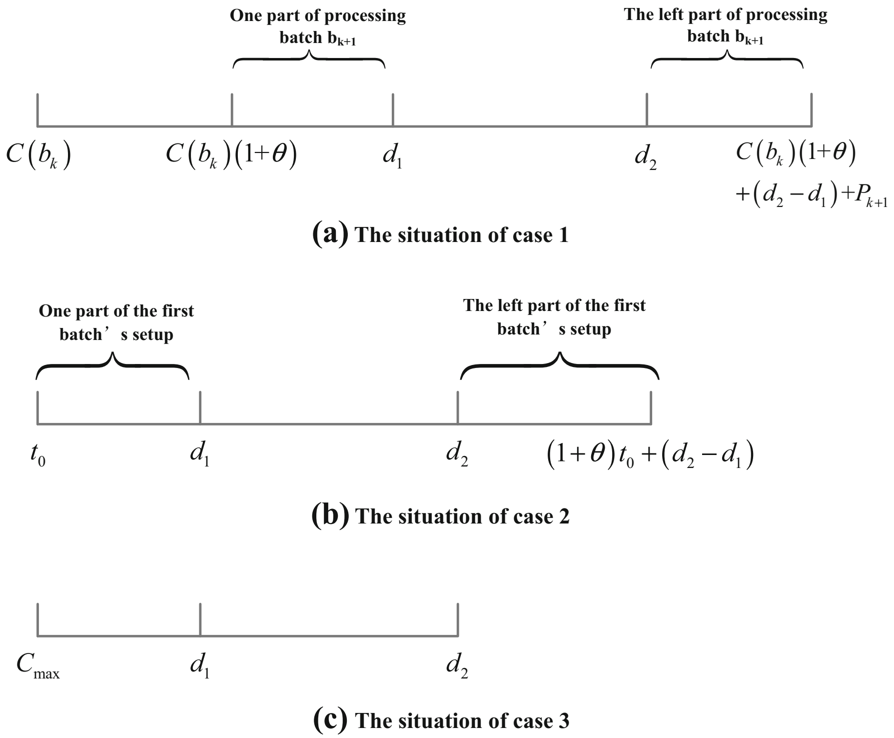

Fig. 1 Three cases for the situation of the machine's non-availability period

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

$$\begin{aligned} C\left(b_{l+1}\right) &= C\left(b_l\right)(1+\theta) + \sum_{i=1+\sum_{j=1}^{l} n_j}^{\sum_{j=1}^{l+1} n_j} p_i\,(a-bi)\,(1+\theta)^{l-k} \\ &= t_0\,(1+\theta)^{l+1} + \sum_{k=1}^{l+1} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i\,(a-bi)\,(1+\theta)^{l-k}, \end{aligned}$$

so Eq. (3) holds for all $1 \le l < f$.

For $l \ge f$, if Eq. (3) holds, then

$$C\left(b_l\right) = t_0\,(1+\theta)^l + \sum_{k=1}^{l} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i\,(a-bi)\,(1+\theta) + (d_2-d_1)\,(1+\theta)^{m-f+1}.$$

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

$$C\left(b_{l+1}\right)=t_0\left(1+\theta\right)^{l+1}+\sum_{k=1}^{l+1}\sum_{i=1+\sum_{j=1}^{k-1}n_j}^{\sum_{j=1}^{k}n_j}p_i\left(a-bi\right)\left(1+\theta\right)+\left(d_2-d_1\right)\left(1+\theta\right)^{m-f}.$$

Thus, Eq. (3) holds for $m=l+1$. Note that $C_{max}\left(\pi\right)=C\left(b_m\right)$, the proof is completed. □

Similar to the Proof of Lemma 1, we have the following lemma

Lemma 2 *For Case 1, given any schedule $\pi=(b_1,b_2,\ldots,b_m)$, with the first batch b_1 starting at time $t_0>0$, if the starting time for the non-availability period is bigger than the starting time of the setup for b_f $(1<f\le m)$and smaller than the completion time of the setup for b_f, and this batch is not the first batch, then the makespan of schedule π is*

$$C_{max}\left(\pi\right)=t_0\left(1+\theta\right)^m+\sum_{k=1}^{m}\sum_{i=1+\sum_{j=1}^{k-1}n_j}^{\sum_{j=1}^{k}n_j}p_i\left(a-bi\right)\left(1+\theta\right)^{m-k}$$
$$+\left(d_2-d_1\right)\left(1+\theta\right)^{m-f+1}.$$

Here we denote the batch set $\left(b_1,b_2,\ldots,b_f\right)$ as batch set A and the batch set $\left(b_{f+1},b_{f+2},\ldots,b_m\right)$ as batch set B, respectively.

Lemma 3 *For Case 1,all jobs should be sequenced in non-decreasing order of p_i in the job sets A and B, respectively.*

Proof We first consider the jobs in B. Here we assume that π and π are an optimal schedule and a job schedule, respectively. The difference of these two schedules is the pairwise interchange of these two jobs J_r and $J_{r+1}(r=1,2,\ldots,n-1)$ in the same batch, that is, $\pi=(W_1,J_r,J_{r+1},W_2)$, $\pi=(W_1,J_{r+1},J_r,W_2)$, where $J_r\in b_p$, $J_{r+1}\in b_p$, and $b_p\subset B$, $n_p\ge 2$, $p=1,2,\ldots,m$. $W_1$and W_2 represent two partial sequences, and W_1 or W_2 may be empty. It is assumed that $p_r\ge p_{r+1}$.

The completion time of b_p in π^* is

$$C\left(b_p\left(\pi^*\right)\right)=t_0\left(1+\theta\right)^p+\sum_{k=1}^{p}\sum_{i=1+\sum_{j=1}^{k-1}n_j}^{\sum_{j=1}^{k}n_j}p_i\left(a-bi\right)\left(1+\theta\right)^{p-k}$$
$$+\left(d_2-d_1\right)\left(1+\theta\right)^{p-f+1}.$$

The completion time of b_p in π is

$$
\begin{aligned}
C\left(b_p(\pi)\right) = t_0(1+\theta)^p + \sum_{k=1}^{p} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} & p_i(a-bi)(1+\theta)^{p-k} \\
& + (d_2-d_1)(1+\theta)^{p-f+1} \\
& - \left[p_r(a-br) + p_{r+1}(a-b(r+1))\right] \\
& + \left[p_{r+1}(a-br) + p_r(a-b(r+1))\right].
\end{aligned}
$$

Then,

$$
\begin{aligned}
C\left(b_p\left(\pi^*\right)\right) - C\left(b_p(\pi)\right) &= \left[p_r(a-br) + p_{r+1}(a-b(r+1))\right] \\
&\quad - \left[p_{r+1}(a-br) + p_r(a-b(r+1))\right] \\
&= (p_r - p_{r+1})\, b.
\end{aligned}
$$

Since we have $p_r > p_{r+1}$ and $b > 0$, it can be derived that

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

which conflicts with the optimal schedule. Hence, $p_r \le p_{r+1}$.

Also, the proof of the case of the jobs in A is similar to that of B, and we omit it. The proof is completed.

Similar to the Proof of Lemma 3, we can obtain the following lemma.

Lemma 4 *For Case 1,the processing time of any job in batch set A is smaller than that in batch set B.*

Based on Lemmas 3 and 4, we can obtain the following corollary.

Corollary 1 *For Case 1, all jobs should be sequenced in non-decreasing order of p_i in the optimal schedule.*

Similar to the Proof of Lemma 3, we can obtain the following two lemmas.

Lemma 5 *For Case 1, for two consecutive batches b_p and b_{p+1} in an optimal schedule, $n_p \ge n_{p+1}$ must hold.*

Lemma 6 *For Case 1, the number of the batches in the optimal schedule is $\left[\frac{n}{c}\right]$.*

Based on Lemmas 5 and 6, we can obtain the following corollary.

Corollary 2 *For Case 1, all batches in an optimal schedule are full, except possibly for the highest indexed one.*

(2) Case 2

Lemma 7 *For Case 2, given any schedule $\pi = (b_1, b_2, \ldots, b_m)$, with the first batch b_1 starting at time $t_0 > 0$, then the makespan of schedule π is*

$$C_{max}(\pi) = t_0 (1+\theta)^m + (d_2 - d_1)(1+\theta)^{m-1} + \sum_{k=1}^{m} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i (a - bi)(1+\theta)^{m-k} \tag{4}$$

Proof Based on the number of the batches, this lemma can be proved by using mathematical induction. Firstly for $m = 1$, there is

$$C(b_1) = t_0 (1+\theta) + (d_2 - d_1) + \sum_{i=1}^{n} p_i (a - bi),$$

so Eq. (4) holds for $m = 1$. Suppose for all $2 \le l \le m-1$, Eq. (4) is satisfied. We have

$$C(b_l) = t_0 (1+\theta)^l + (d_2 - d_1)(1+\theta)^{l-1} + \sum_{k=1}^{l} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i (a - bi)(1+\theta)^{l-k}.$$

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

$$\begin{aligned} C(b_{l+1}) &= (1+\theta)\left[t_0 (1+\theta)^l + (d_2 - d_1)(1+\theta)^{l-1} + \sum_{k=1}^{l} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i (a - bi)(1+\theta)^{l-k}\right] \\ &\quad + \sum_{i=1+\sum_{j=1}^{l} n_j}^{\sum_{j=1}^{l+1} n_j} p_i (a - bi) \\ &= t_0 (1+\theta)^{l+1} + (d_2 - d_1)(1+\theta)^{l} + \sum_{k=1}^{l+1} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i (a - bi)(1+\theta)^{l-k} \end{aligned}$$

Hence, Eq. (4) holds for $m = l+1$. Note that $C_{max}(\pi) = C(b_m)$, the lemma is proved by the induction. □

Lemma 8 *For Case 2, the jobs in the same batch should be sequenced in non-decreasing order of p_i in an optimal schedule.*

Proof Here we assume that π^* and π are an optimal schedule and a job schedule, respectively. The difference of these two schedules is the pairwise interchange of these two jobs J_r and J_{r+1} $(r = 1, 2, \ldots, n-1)$ in the same batch, 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. It is assumed that $p_r \geq p_{r+1}$.

The completion time of b_p in π^* is

$$C\left(b_p\left(\pi^*\right)\right) = t_0\left(1+\theta\right)^p + \left(d_2 - d_1\right)\left(1+\theta\right)^{p-1} + \sum_{k=1}^{p} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i\left(a - bi\right)\left(1+\theta\right)^{p-k},$$

The completion time of b_p in π is

$$\begin{aligned} C\left(b_p\left(\pi\right)\right) = {} & t_0\left(1+\theta\right)^p + \left(d_2 - d_1\right)\left(1+\theta\right)^{p-1} \\ & + \sum_{k=1}^{p} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i\left(a - bi\right)\left(1+\theta\right)^{p-k} \\ & - \left[p_r\left(a - br\right) + p_{r+1}\left(a - b\left(r+1\right)\right)\right] \\ & + \left[p_{r+1}\left(a - br\right) + p_r\left(a - b\left(r+1\right)\right)\right]. \end{aligned}$$

Then,

$$\begin{aligned} & C\left(b_p\left(\pi^*\right)\right) - C\left(b_p\left(\pi\right)\right) \\ & \quad = \left[p_r\left(a - br\right) + p_{r+1}\left(a - b\left(r+1\right)\right)\right] \\ & \qquad - \left[p_{r+1}\left(a - br\right) + p_r\left(a - b\left(r+1\right)\right)\right] \\ & \quad = \left(p_r - p_{r+1}\right) b. \end{aligned}$$

Since we have $p_r > p_{r+1}$ and $b > 0$, it can be derived that

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

which conflicts with the optimal schedule. Hence, $p_r \leq p_{r+1}$. The proof is completed. □

Lemma 9 *For Case 2, for two jobs $J_u \in b_p$ and $J_v \in b_{p+1}$ from two consecutive batches, in an optimal schedule $p_u \leq p_v$must hold.*

Proof Here we assume that π^* and π are an optimal schedule and a job schedule, respectively. The two schedules' difference is the pairwise interchange of these two jobs J_u and J_v (where J_u and J_v are in the u-th and **v**-th positions, and $u < v$), that is, $\pi = \left(W_1, b_p, b_{p+1}, W_2\right)$, $\pi = \left(W_1, \left(b_p / \left\{J_u\right\}\right) \cup \left\{J_v\right\}, \left(b_{p+1} / \left\{J_v\right\}\right) \cup \left\{J_u\right\}, W_2\right)$. $W_1$and W_2 represent two partial sequences, and W_1 or W_2 may be empty. It is assumed that $p_u > p_v$.

The completion time of b_{p+1} in π is

$$C\left(b_{p+1}\left(\pi^*\right)\right) = t_0\left(1+\theta\right)^{p+1} + \left(d_2 - d_1\right)\left(1+\theta\right)^p + \sum_{k=1}^{p+1} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i\left(a - bi\right)\left(1+\theta\right)^{p+1-k}.$$

The completion time of b_{p+1} in π is

$$\begin{aligned} C\left(b_{p+1}\left(\pi\right)\right) = {} & t_0\left(1+\theta\right)^{p+1} + \left(d_2 - d_1\right)\left(1+\theta\right)^p \\ & + \sum_{k=1}^{p+1} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i\left(a - bi\right)\left(1+\theta\right)^{p+1-k} \\ & - \left[p_u\left(a - bu\right)\left(1+\theta\right) + p_v\left(a - bv\right)\right] \\ & + \left[p_v\left(a - bu\right)\left(1+\theta\right) + p_u\left(a - bv\right)\right]. \end{aligned}$$

Then,

$$\begin{aligned} C\left(b_p\left(\pi^*\right)\right) - C\left(b_p\left(\pi\right)\right) &= \left(p_u - p_v\right)\left(a - bu\right)\left(1+\theta\right) - \left(p_u - p_v\right)\left(a - bv\right) \\ &= \left(p_u - p_v\right)\left[\theta\left(a - bu\right) + b\left(v - u\right)\right]. \end{aligned}$$

Hence, it can be derived that

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

which conflicts with the optimal schedule. Thus, $p_u \leq p_v$. The proof is completed. □

Based on Lemmas 8 and 9, we have the following Corollary 3.

Corollary 3 *For Case 2, all jobs should be sequenced in non-decreasing order of p_i in the optimal schedule.*

Then, we have the following lemma similar to Case 1.

Lemma 10 *For Case 2, the number of the batches in the optimal schedule should be $\left\lceil \frac{n}{c} \right\rceil$. All batches are full in an optimal schedule, except possibly for the highest indexed one.*

(3) Case 3

Lemma 11 *For Case 3, given any schedule $\pi = (b_1, b_2, \ldots, b_m)$, with the first batch b_1 starting at time $t_0 > 0$, then the makespan of schedule π is*

$$C_{max}\left(\pi\right) = t_0\left(1+\theta\right)^m + \sum_{k=1}^{m} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i\left(a - bi\right)\left(1+\theta\right)^{m-k}.$$

Similar to the Proof of Lemmas in Cases 1 and 2, we have the following lemma.

Lemma 12 *For Case 3, all jobs should be sequenced in non-decreasing order of* p_i. *The number of the batches in the optimal schedule should be* $\left[\frac{n}{c}\right]$. *All batches in an optimal schedule are full, except possibly for the highest indexed one.*

Based on all lemmas and corollaries, a heuristic algorithm is proposed to the solve this problem, and it is described as follows:

Algorithm SPT-FB (Shortest processing time and full batch first)

Step 1. Sort all jobs in non-decreasing order of their normal processing time p_i, i.e., $p_1 \leq p_2 \leq \cdots \leq p_n$, and obtain a job list.

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. Schedule all jobs at the time t_0 as the generated batches' sequence in step 2. Stop the processing at the time d_1, and continue the processing at the time d_2 in the previous order until all jobs are processing

Theorem 1 *For the problem* $1\left|s-batch, r-a, p_{ir}=p_i\left(a-br\right), s=\theta t\right| C_{max}$, *an optimal schedule can be obtained by Algorithm SPT-FB in* $O\left(n\log n\right)$ *time. If all job are sequenced in non-increasing order of* p_i $\left(i=1,2\ldots,n\right)$, *then the results of the optimal makespan are as follows*:

1. *when* $t_0 \leq d_1 \leq (1+\theta)\, t_0$,

$$C_{max} = t_0\left(1+\theta\right)^{\left[\frac{n}{c}\right]} + \left(d_2-d_1\right)\left(1+\theta\right)^{\left[\frac{n}{c}\right]} + \sum_{k=1}^{\left[\frac{n}{c}\right]} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i\left(a-bi\right)\left(1+\theta\right)^{\left[\frac{n}{c}\right]-k}.$$

2. *when* $(1+\theta)\, t_0 \leq d_1 \leq t_0\left(1+\theta\right)^{\left[\frac{n}{c}\right]} + \sum_{k=1}^{\left[\frac{n}{c}\right]} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i\left(a-bi\right)\left(1+\theta\right)^{\left[\frac{n}{c}\right]-k}$,

$$C_{max} = t_0\left(1+\theta\right)_{\left[\frac{n}{c}\right]} + \sum_{k=1}^{\left[\frac{n}{c}\right]} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i\left(a-bi\right)\left(1+\theta\right)^{\left[\frac{n}{c}\right]-k} + \left(d_2-d_1\right)\left(1+\theta\right)^{\left[\frac{n}{c}\right]-f+1},$$

where the non-availability period happens during the setup operation or while processing b_f.

Table 1 An illustrative example

Job	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}
Processing time	12	30	21	8	5	2	16	9	7	35

3. *when* $t_0(1+\theta)^{\left[\frac{n}{c}\right]} + \sum_{k=1}^{\left[\frac{n}{c}\right]} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i (a-bi)(1+\theta)^{\left[\frac{n}{c}\right]-k} \le d_1$,

$$C_{max} = t_0(1+\theta)^{\left[\frac{n}{c}\right]} + \sum_{k=1}^{\left[\frac{n}{c}\right]} \sum_{i=1+\sum_{j=1}^{k-1} n_j}^{\sum_{j=1}^{k} n_j} p_i (a-bi)(1+\theta)^{\left[\frac{n}{c}\right]-k}.$$

Proof Combining Lemmas 1–12 and Corollaries 1 and 2, we can obtain that Algorithm SPT-FB can generate an optimal schedule for the problem $1\left|s-batch, r-a, p_{ir}=p_i(a-br), s=\theta t\right| C_{max}$.

And the result of the optimal makespan can be derived as the above three cases. The complexity of Step 1 is $O(n\log n)$, the complexity of Step 2 is $O(n)$, and the complexity of Step 3 is $O(1)$. Thus, the total complexity of this proposed algorithm is $O(n\log n)$. □

In addition, we demonstrate the result of Theorem 1 in the following example.

Example 1 Assume $n=10$, $c=3$, $t_0=2$, $\theta=0.5$, $a=10$, $b=0.1$, $d_1=800$, $d_2=900$, the normal processing times of jobs are given in Table 1.

Solution. According to Algorithm SPT-FB, we solve Example 1 as follows:

Step 1. Sort all jobs in non-decreasing order of their normal processing time, and the optimal job sequence is $J_6 \to J_5 \to J_9 \to J_4 \to J_8 \to J_1 \to J_7 \to J_3 \to J_2 \to J_{10}$.

Step 2. Batch the jobs as the rule of full batch first, and the optimal batches are $\{J_6, J_5, J_9\}$, $\{J_4, J_8, J_1\}$, $\{J_7, J_3, J_2\}$, $\{J_{10}\}$.

Thus, in the optimal schedule, we have

$$\begin{aligned}
C(b_1) &= 2\times(1+0.5)+[2\times(10-1\times 0.1)+5\times(10-2\times 0.1)+7\times(10-3\times 0.1)]\\
&= 139.7,\\
C(b_2) &= 139.7\times(1+0.5)+[8\times(10-4\times 0.1)+9\times(10-5\times 0.1)+12\times(10-6\times 0.1)]\\
&= 484.65,\\
C(b_3) &= 484.65\times(1+0.5)+[16\times(10-7\times 0.1)+21\times(10-8\times 0.1)+30\times(10-9\times 0.1)]\\
&\quad +(900-800) = 1441.975,\\
C(b_4) &= 1441.975\times(1+0.5)+35\times(10-10\times 0.1) = 2477.9625.
\end{aligned}$$

The optimal makespan is 2477.9625.

4 Conclusion

In this paper we investigate the problem of scheduling the jobs with learning effect on a single serial-batching machine with an availability constraint. The objective is to find an optimal schedule that minimizes the makespan. Three different cases of this scheduling problem are considered, and their properties are derived and investigated. Based on these properties, an optimization algorithm is proposed to solve this problem.

Further research might focus on other objectives, such as the total completion time and total tardiness. It is also interesting to extend this problem to parallel machines scheduling problem or flowshop problem.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Nos. 71601065, 71231004, 71501058, 71131002, 71471052, 71521001, 71571058), the Humanities and Social Sciences Foundation of the Chinese Ministry of Education (No. 15YJC630097), Anhui Province Natural Science Foundation (No. 1608085QG167), and the Fundamental Research Funds for the Central Universities (Nos. JZ2016HGTA0709, JZ2016HGTB0727). 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. Wright, T.P.: Factors affecting the cost of airplanes. J. Aeronaut. Sci. **3**, 122–128 (1936)
3. Cheng, T.C.E., Wang, G.: Single machine scheduling with learning effect considerations. Ann. Oper. Res. **98**(1), 273–290 (2000)
4. Wang, J.-B., Wang, J.-J.: Single machine scheduling with sum-of-logarithm-processing-times based and position based learning effects. Optim. Lett. **8**(3), 971–982 (2012)
5. Lee, W.-C.: Single-machine scheduling with past-sequence-dependent setup times and general effects of deterioration and learning. Optim. Lett. **8**(1), 135–144 (2014)
6. Cheng, T.C.E., Kuo, W.-H., Yang, D.-L.: Scheduling with a position-weighted learning effect. Optim. Lett. **8**(1), 293–306 (2014)
7. Wang, J.-B., Wang, J.-J.: Flowshop scheduling with a general exponential learning effect. Comput. Oper. Res. **43**, 292–308 (2014)
8. Luo, K., Zhang, X.: A simple proof technique for scheduling models with learning effects. Optim. Lett. **9**(7), 1411–1420 (2015)
9. Li, G., Luo, M.-L., Zhang, W.-J., Wang, X.-Y.: Single-machine due-window assignment scheduling based on common flow allowance, learning effect and resource allocation. Int. J. Prod. Res. **53**(4), 1228–1241 (2015)
10. Lee, C.Y.: Machine scheduling with an availability constraint. J. Global Optim. **9**(3), 395–416 (1996)
11. Lee, C.Y.: Two-machine flowshop scheduling with availability constraints. Eur. J. Oper. Res. **114**(2), 420–429 (1999)
12. Zhong, X., Ou, J., Wang, G.: Order acceptance and scheduling with machine availability constraints. Eur. J. Oper. Res. **232**(3), 435–441 (2014)
13. Kacem, I., Kellerer, H., Lanuel, Y.: Approximation algorithms for maximizing the weighted number of early jobs on a single machine with non-availability intervals. J. Comb. Optim. **30**(3), 403–412 (2015)
14. Mor, B., Mosheiov, G.: Heuristics for scheduling problems with an unavailability constraint and position-dependent processing times. Comput. Ind. Eng. **62**(4), 908–916 (2012)
15. Wu, C.-C., Lee, W.-C.: A note on single-machine scheduling with learning effect and an availability constraint. Int. J. Adv. Manuf. Technol. **33**(5), 540–544 (2007)
16. 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)
17. 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)

18. 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)
19. Pei, J., Liu, X., Pardalos, P.M., Fan, W., Yang, S.: Scheduling deteriorating jobs on a single serial-batching machine with multiple job types and sequence-dependent setup times. Ann. Oper. Res. (2015). doi:10.1007/s10479-015-1824-6
20. Pei, J., Liu, X., Pardalos, P.M., Migdalas, A., Yang, S.: Serial-batching scheduling with time-dependent setup time and effects of deterioration and learning on a single-machine. J. Glob. Optim. (2015). doi:10.1007/s10898-015-0320-5
21. 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)
22. 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)
23. Huang, X., Wang, M.Z., Wang, J.B.: Single-machine group scheduling with both learning effects and deteriorating jobs. Comput. Ind. Eng. **60**(4), 750–754 (2011)
24. Pei, J., Liu, X., Pardalos, P.M., Fan, W., Yang, S.: Single machine serial-batching scheduling with independent setup time and deteriorating job processing times. Optim. Lett. **9**(1), 91–104 (2015)
25. Wang, J.-B., Xia, Z.-Q.: Flow-shop scheduling with a learning effect. J. Oper. Res. Soc. **56**(11), 1325–1330 (2005)
26. Xuan, H., Tang, L.X.: Scheduling a hybrid flowshop with batch production at the last stage. Comput. Oper. Res. **34**(9), 2718–2733 (2007)
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. Wu, C.-C., Lee, W.-C.: Scheduling linear deteriorating jobs to minimize makespan with an availability constraint on a single machine. Inform. Process. Lett. **87**(2), 89–93 (2003)
29. Gawiejnowicz, S., Kononov, A.: Complexity and approximability of scheduling resumable proportionally deteriorating jobs. Eur. J. Oper. Res. **200**(1), 305–308 (2010)
30. 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. Discrete Math. **5**, 287–326 (1979)