Journal of the Operations Research Society of China
https://doi.org/10.1007/s40305-023-00480-6

# A Best Possible Online Algorithm For Parallel-Batch Scheduling with Kind Release Times and Job Compatibilities

Li-Yun Miao[1] · Ji Tian[1] · Ru-Yan Fu[1]

Received: 25 May 2022 / Revised: 25 February 2023 / Accepted: 26 February 2023

**Abstract**
We investigate the problem of the online scheduling with kind release times and job compatibilities on a single unbounded parallel-batch machine to minimize makespan. The kind release times (KRT) means that under the online setting no jobs can be released when the machine is busy. What associated with each job $J_j$ are its normal processing time $p_j$ and release time $r_j$. Two jobs $J_i$ and $J_j$ are called compatible if $\max\{p_i, p_j\} \leqslant (1+a)\min\{p_i, p_j\}$, where $a$ is a given positive constant. Compatible jobs could be processed in the same batch. We derive a best possible online algorithm with a competitive ratio of $1+\sqrt{\lambda^2-\lambda+1}-\lambda$, where $\lambda=\frac{a}{1+a}$.

**Keywords** Online scheduling · Kind release time · Compatibility · Batch machine · Competitive ratio

**Mathematics Subject Classification** 90B35

## 1 Introduction

Online scheduling problems have been extensively studied in the last three decades. In this paper, "online" means that all jobs arrive over time. Each job is completely unknown until its release time. Here, we consider an onlinescheduling problem with

This work was supported by the National Natural Science Foundation of China (No. 61573362).

✉ Ji Tian
jitian@cumt.edu.cn

Li-Yun Miao
ts19080036a31@cumt.edu.cn

Ru-Yan Fu
furuyan@cumt.edu.cn

[1] School of Mathematics, China University of Mining and Technology, Xuzhou 221116, Jiangsu, China

Published online: 04 April 2023

kind release times (KRT). KRT means that under the online setting no jobs can be released if the machine is busy. There are differences between the KRT environment and the traditional online setting. In the former, an online algorithm deals with job instances depending on its own strategy, while there is no such limitation in the latter model.

As in the work of traditional online scheduling, under KRT assumption, the quality of an online algorithm $H$ is also measured by its competitive ratio

$$\rho_H = \sup\{H(L)/\mathrm{OPT(L)}\},$$

where $L$ is any job instance depending on the online algorithm $H$, $H(L)$ denotes the objective value of the schedule obtained by algorithm $H$ for $L$, and OPT(L) denotes the corresponding objective value of some offline optimal schedule. With regard to the above definition, the following two points should be noted. One is that $L$ is an instance depending on the online algorithm $H$, not arbitrary. The other is that the KRT assumption is a requirement of online setting, which is not taken into account when discussing the offline optimal value OPT(L).

Parallel-batch scheduling problems have been widely and deeply researched in recent years. An unbounded parallel batch machine can handle arbitrary number of jobs simultaneously in a batch. All jobs in a batch start and finish at the same time, and they have the same actual processing time, which is usually defined as the longest normal processing time of all jobs in it. In the literature on unbounded parallel-batch scheduling problems, most researchers commonly assume that the actual job processing time of a job has no upper bound (see [1–11]). However, an overlength actual processing time on a job may cause defects in practical production [12]. In this paper, we assume that a job's actual processing time cannot exceed $1 + a$ times its normal value, where $a$ is a given positive number.

Now we state our problem in detail: There are $n$ jobs $J_1, J_2, \ldots, J_n$ to be scheduled on an unbounded batch machine. Each job $J_j$ has a normal processing time $p_j$ and a release time $r_j$. The characteristics of each job become known at its release time. Jobs have a kind release time (KRT) in online setting and satisfy compatibility constraints. Two jobs $J_i$ and $J_j$ are compatible if $\max\{p_i, p_j\} \leqslant (1 + a)\min\{p_i, p_j\}$. Here, the job compatibility relations can be described by an specific interval compatibility graph $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$ is the vertex set, and there is an edge $(i, j) \in E$ if and only if jobs $J_i$ and $J_j$ are compatible (i.e., $[p_i, (1+a)p_i] \cap [p_j, (1+a)p_j] \neq \varnothing$). The batch machine can handle any number of jobs in a batch as long as these jobs are pairwise compatible. All jobs in a batch $B$ have the same actual processing time, which is equal to $\max\{p_j : J_j \in B\}$. The goal is to find a feasible schedule of the jobs to minimize the maximum completion time, or makespan. The problem can be denoted by $1|$online, KRT-$r_j$, p-batch, G $= a$-INT$|C_{\max}$, where KRT-$r_j$ denotes the arrival of jobs follows KRT assumption in online setting, and G $= a$-INT denotes the job compatibility relations.

In the literature, most of the work on batch scheduling problems with job compatibilities characterized by compatibility graphs are about offline scheduling (see [12–15]). In online setting, for problem $1|$online, $r_j$, p-batch, G $= a$-INT$|C_{\max}$, Li et al. [16] provided a class of online algorithms with competitive ratio of 2. Fu et al. [17]

designed a best possible online algorithm with a competitive ratio of $1+\frac{\sqrt{\lambda^2+4}-\lambda}{2}$, where $\lambda=\frac{a}{1+a}$.

There is not a lot of the literature study online scheduling problems with KRT assumption. For problem $P2|\text{online, KRT-}r_j|C_{\max}$, Li and Yuan [18] showed that LPT rule is best possible and has a competitive ratio of $\frac{5}{4}$. For problem $1|\text{online, KRT-}r_j\text{, p-batch, } b=\infty\text{, } f\text{-family}|C_{\max}$, where $f$-family denotes the number of job families (any two jobs belonging to distinct families cannot be processed in the same batch), Li et al. [19] established a best online algorithm with competitive ratio of $1+\frac{\sqrt{f^2-f+1}-1}{f}$.

In this paper, we will study the online scheduling problem $1|\text{online, KRT-}r_j$, p-batch, $\text{G}=a\text{-INT}|C_{\max}$. It is organized as follows: In Sects. 2 and 3, we will introduce our online algorithm and show that its competitive ratio is not greater than $1+\sqrt{\lambda^2-\lambda+1}-\lambda$. In Sect. 4, by deriving a lower bound on the competitive ratio of any online algorithm with KRT assumption, we will prove that our online algorithm is best possible.

## 2 An Online Algorithm

For the offline problem $1|\text{p-batch, G}=a\text{-INT}|C_{\max}$, Bellanger et al. [12] presented a BCLPT rule to solve it in $O(n\log n)$ time. Given that this rule will be a sub-procedure of our online algorithm, we describe it as follows:

**BCLPT rule**
**Step 1** Index all jobs in a job set $\mathcal{J}$ by nonincreasing normal processing time, i.e., $p_1 \geqslant p_2 \geqslant \cdots \geqslant p_n$.
**Step 2** Do the following:

- Put jobs $J_1, J_2, \cdots, J_{i_1}$ in batch $B_1$ until a certain $i_1$ satisfies that $(1+a)p_{i_1+1} < p_1$. If such an $i_1$ does not exist, set $B_1=\mathcal{J}$ and stop.
- Put jobs $J_{i_1+1}, J_{i_1+2}, \cdots, J_{i_2}$ in batch $B_2$ until a certain $i_2$ satisfies that $(1+a)p_{i_2+1} < p_{i_1+1}$. If such an $i_2$ does not exist, set $B_2=\mathcal{J}\backslash B_1$ and stop.
- Continue the above procedure until job $J_n$ is placed in $B_k$ for some $k$ and stop.

**Step 3** Sequence the batches $B_1, B_2, \cdots, B_k$ in an arbitrary order.

For a job set, the BCLPT rule may produce more than one batches, and we call them BCLPT batches. To facilitate the description of our online algorithm, we will use the following notation.

$J(B)$ denotes the representative job in a BCLPT batch $B$ which has the longest normal processing time and arrives as late as possible. Given that the batch capacity $b=\infty$, we suppose that there is only one such job in a batch. Its normal processing time and release time are denoted by $p(B)$ and $r(B)$, respectively;

$U(t)$ is the set containing all jobs that have arrived at or before time $t$ and that have not been scheduled by time $t$;

$\mathcal{A}(t)$ is the set of all BCLPT batches produced by applying BCLPT rule on $U(t)$;

$V(t)$ is the set of all the representative jobs in BCLPT batches of $\mathcal{A}(t)$;

$r(X)$ denotes the minimum release time of all jobs in a job set $X$;

$P(X)$ denotes the total normal processing time of all jobs in a job set $X$.

Now we begin to describe our online algorithm. The main difference between our algorithm and the algorithm presented by Fu et al. [17] is the value of parameter $\beta$. Here, $\beta = \sqrt{\lambda^2 - \lambda + 1} - \lambda$, where $\lambda = \frac{a}{1+a}$.

**Algorithm** $H$

**Step 0** Set $t = 0$.
**Step 1** If $U(t) = \emptyset$, go to Step 3. Otherwise, call BCLPT rule on $U(t)$. Determine $\mathcal{A}(t)$ and $V(t)$.
**Step 2** If $t \geq \beta \cdot P(V(t))$, choose the longest waiting batch in $\mathcal{A}(t)$ and process it immediately; reset $t$ to the completion time of this batch. Otherwise, reset $t = \beta \cdot P(V(t))$. Go to Step 1.
**Step 3** Reset $t$ to the next time (if any) at which a new job arrives. Go to Step 1.

In online algorithm $H$, at each decision time $t$, we assume that $\mathcal{A}(t) = \{B_1, B_2, \cdots, B_{m_t}\}$, where $m_t$ is the number of BCLPT batches in $\mathcal{A}(t)$. Then $V(t) = \{J(B_1), J(B_2), \cdots, J(B_{m_t})\}$, $r(V(t)) = \min\{r(B_i) : 1 \leqslant i \leqslant m_t\}$ and $P(V(t)) = \sum_{i=1}^{m_t} p(B_i)$. Without loss of generality, we suppose that $p(B_1) > p(B_2) > \cdots > p(B_{m_t})$. According to the algorithm $H$ and BCLPT rule, we obtain following Lemma 1.

**Lemma 1** *Suppose that $B$ is a batch starting at time $t$ in the schedule generated by algorithm $H$. Then the following properties hold*:

(a) *$B = B_1$ is the longest waiting BCLPT batch in $\mathcal{A}(t)$ at time $t$;*
(b) *Any two jobs in $V(t)$ are incompatible, i.e., $(1+a)p(B_{i+1}) < p(B_i)$ for $1 \leqslant i \leqslant m_t - 1$;*
(c) *$t \geqslant \beta \cdot P(V(t))$. Furthermore, if there is an idle time directly before time $t$, then $t = \max\{r(V(t)), \beta \cdot P(V(t))\}$.*

## 3 Analysis of Algorithm *H*

In this section we show that the competitive ratio of algorithm $H$ is not greater than $1 + \sqrt{\lambda^2 - \lambda + 1} - \lambda$. Here, $\beta = \sqrt{\lambda^2 - \lambda + 1} - \lambda$ satisfies the equation $x^2 + 2\lambda x + (\lambda - 1) = 0$, where $\lambda = \frac{a}{1+a}$. Note that the above equation is equivalent to the equation $(1+a)x^2 + 2ax = 1$. Then $\beta$ is also the positive root of the later, and we have

$$\frac{1}{\beta(1+a) + 2a} = \beta, \tag{1}$$

which will be repeatedly used in the analysis to be presented in this section. Given an input job instance, let $\sigma$ be the schedule produced by algorithm $H$, and $\pi$ be an offline optimal schedule. We also use $C_{\mathrm{on}}$ and $C_{\mathrm{opt}}$ to denote the makespan of the schedules $\sigma$ and $\pi$, respectively.

Our proof is by contradiction. Suppose the competitive ratio of algorithm $H$ exceeds $1 + \beta$, then there exists a job instance $\mathcal{I}$, which we call counter-example, such that

$$C_{\text{on}}/C_{\text{opt}} > 1 + \beta. \tag{2}$$

Let $r_l$ be the release time of the last job in instance $\mathcal{I}$. Suppose that $\tau \geqslant r_l$ be the minimum starting time such that the machine is busy in the interval $[\tau, C_{\text{on}}]$. Without causing confusion, let $\mathcal{A}(\tau) = \{B_1, B_2, \cdots, B_k\}$ and $V(\tau) = \{J(B_1), J(B_2), \cdots, J(B_k)\}$. According to the definition of $\tau$ and algorithm $H$, we know that all BCLPT batches in $\mathcal{A}(\tau)$ are consecutively processed from time $\tau$ to $C_{\text{on}}$. Then we have

$$C_{\text{on}} = \tau + P(V(\tau)) = \tau + \sum_{i=1}^{k} p(B_i). \tag{3}$$

Furthermore, by Lemma 1(b), we obtain the following claim.

**Claim 1** *All jobs in $V(\tau)$ are pairwise incompatible, and $(1+a)p(B_{i+1}) < p(B_i)$ for $1 \leqslant i \leqslant k-1$.*

Claim 1 implies that all jobs in $V(\tau)$ should be processed independently in $k$ distinct batches in any feasible schedule. Then we have

$$C_{\text{opt}} \geqslant r(V(\tau)) + P(V(\tau)) = r(V(\tau)) + \sum_{i=1}^{k} p(B_i). \tag{4}$$

**Claim 2** *There does not exist an idle time immediately before time $\tau$ in $\sigma$.*

***Proof*** Suppose to the contrary that $\sigma$ does not have this form. Then, by Lemma 1($c$), we have $\tau = \max\{r(V(\tau)),\ \beta \cdot P(V(\tau))\}$. From (3) and (4), we get $C_{\text{on}} - C_{\text{opt}} \leqslant \tau - r(V(\tau)) \leqslant \max\{0,\ \beta \cdot P(V(\tau)) - r(V(\tau))\} \leqslant \beta \cdot P(V(\tau)) \leqslant \beta C_{\text{opt}}$. This contradicts (2). The claim follows.

Claim 2 implies that $\tau$ is the completion time of some batch, say, $B_0$. Let $\tau_0$ be the starting time of batch $B_0$ in $\sigma$. Then $\tau = \tau_0 + p(B_0)$. By the definition of $r_l$ and the choice of $\tau$, we can deduce that $r_l > \tau_0$. Given that the machine is busy from time $\tau_0$ to $\tau$, then the KRT assumption implies that no jobs arrive in the interval $(\tau_0, \tau)$. Note that $r_l \leqslant \tau$. Then $r_l = \tau$, and we can divide $V(\tau)$ into two sets, say, $\mathcal{Q}$ and $\mathcal{R}$, such that

$$\mathcal{Q} = \{J(B_i) : r(B_i) \leqslant \tau_0, 1 \leqslant i \leqslant k\} \text{ and } \mathcal{R} = \{J(B_i) : r(B_i) = \tau, 1 \leqslant i \leqslant k\}.$$

Then $P(V(\tau)) = P(\mathcal{Q}) + P(\mathcal{R})$, and by (3), we have

$$C_{\text{on}} = \tau_0 + p(B_0) + P(\mathcal{Q}) + P(\mathcal{R}). \tag{5}$$

Since $r(\mathcal{R}) = \tau = \tau_0 + p(B_0)$, we have $C_{\text{opt}} \geqslant r(\mathcal{R}) + P(\mathcal{R}) = \tau_0 + p(B_0) + P(\mathcal{R})$, where the case $\mathcal{R} = \varnothing$ is included since $C_{\text{opt}} \geqslant r_l = \tau = \tau_0 + p(B_0)$. Then, combining (5), we have

$$C_{\text{on}} - C_{\text{opt}} \leqslant P(\mathcal{Q}). \tag{6}$$

From inequalities (2) and (6), we have $P(\mathcal{Q}) > \beta C_{\text{opt}}$, and so $\mathcal{Q} \neq \varnothing$. Suppose that $\mathcal{Q} = \{J(B_{i_1}), J(B_{i_2}), \cdots, J(B_{i_q})\}$ such that $p(B_{i_{j+1}}) < p(B_{i_j})$ for $1 \leqslant j \leqslant q-1$. Then we have following Observation 1.

**Observation 1** *(a) $(\mathcal{Q} \cup J(B_0)) \subseteq U(\tau_0)$; (b) Every two jobs in $\mathcal{Q} \cup J(B_0)$ are incompatible; furthermore, $(1+a)p(B_{i_{j+1}}) < p(B_{i_j})$ for $1 \leqslant j \leqslant q-1$, and $(1+a)p(B_{i_1}) < p(B_0)$; (c) $P(V(\tau_0)) \geqslant p(B_0) + P(\mathcal{Q})$.*

***Proof*** The definitions of set $\mathcal{Q}$ and batch $B_0$ imply $(a)$ holds. Since $\mathcal{Q} \subseteq V(\tau)$, by Claim 1, we have that any pair of jobs from $\mathcal{Q}$ are incompatible and $(1+a)p(B_{i_{j+1}}) < p(B_{i_j})$ for $1 \leqslant j \leqslant q-1$. From Lemma 1$(a)$, we know that $B_0$ is the longest waiting BCLPT batch in $\mathcal{A}(\tau_0)$. Note that $\mathcal{Q} \cap B_0 = \varnothing$. Then, by the BCLPT rule, any job in $\mathcal{Q}$ is incompatible with $J(B_0)$, and so $(1+a)p(B_{i_1}) < p(B_0)$. Hence, $(b)$ holds. $(a)$ and $(b)$ together imply that the jobs from $\mathcal{Q} \cup J(B_0)$ are in distinct BCLPT batches at time $\tau_0$. Note that $V(\tau_0)$ is the set of all the representative jobs of BCLPT batches in $\mathcal{A}(\tau_0)$. Then $P(V(\tau_0)) \geqslant p(B_0) + P(\mathcal{Q})$. Hence, $(c)$ holds.

From Observation 1$(b)$, we can deduce that

$$P(\mathcal{Q}) = \sum_{j=1}^{q} p(B_{i_j}) < \left(\frac{1}{1+a} + \frac{1}{(1+a)^2} + \cdots + \frac{1}{(1+a)^q}\right) p(B_0) < \frac{1}{a} p(B_0). \tag{7}$$

**Claim 3** *The starting time of $J(B_0)$ is smaller than $\tau$ in $\pi$.*

***Proof*** Suppose to the contrary that $\pi$ does not have this form. Then $C_{\text{opt}} \geqslant \tau + p(B_0) = \tau_0 + 2p(B_0)$. Furthermore, by Lemma 1$(c)$ and Observation 1$(c)$, we obtain $C_{\text{opt}} \geqslant \beta \cdot (p(B_0) + P(\mathcal{Q})) + 2p(B_0)$. Then, together with inequalities (6) and (7), we have

$$\begin{aligned} \frac{C_{\text{on}} - C_{\text{opt}}}{C_{\text{opt}}} &\leqslant \frac{P(\mathcal{Q})}{\beta \cdot (p(B_0) + P(\mathcal{Q})) + 2p(B_0)} < \frac{\frac{1}{a} p(B_0)}{\beta \cdot (p(B_0) + \frac{1}{a} p(B_0)) + 2p(B_0)} \\ &= \frac{1}{\beta(1+a) + 2a} = \beta. \end{aligned}$$

The last equality follows from (1). This contradicts (2) again. The claim follows.

**Claim 4** *Every two jobs from $\mathcal{Q} \cup \mathcal{R} \cup J(B_0)$ are processed in distinct batches in $\pi$.*

***Proof*** The fact $r(\mathcal{R}) = \tau$ reveals that the starting time of each job from $\mathcal{R}$ is not smaller than $\tau$ in $\pi$. Then, by Claim 3, any job belonging to $\mathcal{R}$ will not be processed in the batch containing $J(B_0)$ in $\pi$. From Claim 1 and Observation 1$(b)$, we know that every two jobs from $\mathcal{Q} \cup \mathcal{R}$ or $\mathcal{Q} \cup J(B_0)$ are incompatible. Therefore, every two jobs from $\mathcal{Q} \cup \mathcal{R} \cup J(B_0)$ cannot be processed in a same batch in $\pi$. The claim follows.

Note that $r(\mathcal{Q}) \leqslant \tau_0 < \tau = r(\mathcal{R})$. From Claim 3 and Claim 4, we have

$$C_{\text{opt}} \geqslant \min\{r(B_0), r(\mathcal{Q})\} + p(B_0) + P(\mathcal{Q}) + P(\mathcal{R}). \tag{8}$$

**Claim 5** *There does not exist an idle time immediately before time* $\tau_0$ *in* $\sigma$.

***Proof*** Suppose to the contrary that $\sigma$ does not have this form. Then, by Lemma 1($c$), we have $\tau_0 = \max\{r(V(\tau_0)),\ \beta \cdot P(V(\tau_0))\}$.

If $\tau_0 = \beta \cdot P(V(\tau_0))$, by inequalities (5) and (8), together with the fact $C_{\text{opt}} \geqslant P(V(\tau_0))$, we have $C_{\text{on}} - C_{\text{opt}} \leqslant \tau_0 \leqslant \beta C_{\text{opt}}$, a contradiction.

If $\tau_0 > \beta \cdot P(V(\tau_0))$, then $\tau_0 = r(V(\tau_0))$. In this case, by the implementation of algorithm $H$, we further deduce that all jobs in $U(\tau_0)$ arrive at time $\tau_0$. From Observation 1($a$), we have $r(B_0) = r(\mathcal{Q}) = \tau_0$. Then, by inequalities (8) and (5), we have that $C_{\text{opt}} \geqslant \tau_0 + p(B_0) + P(\mathcal{Q}) + P(\mathcal{R}) = C_{\text{on}}$, a contradiction. The claim follows.

Claim 5 implies that $\tau_0$ is the completion time of some batch, say, $B^*$. Let $\tau^*$ be the starting time of batch $B^*$ in $\sigma$. Then $\tau_0 = \tau^* + p(B^*)$, and by (5), we have

$$C_{\text{on}} = \tau^* + p(B^*) + p(B_0) + P(\mathcal{Q}) + P(\mathcal{R}). \tag{9}$$

Recall that $r_l = \tau = \tau^* + p(B^*) + p(B_0)$. From lemma 1($c$), we have

$$C_{\text{opt}} \geqslant r_l \geqslant \beta \cdot P(V(\tau^*)) + p(B^*) + p(B_0). \tag{10}$$

**Claim 6** $r(\mathcal{Q}) \leqslant \tau^*$.

***Proof*** Firstly, we assert $\min\{r(B_0), r(\mathcal{Q})\} \leqslant \tau^*$. Otherwise, $\min\{r(B_0), r(\mathcal{Q})\} > \tau^*$. Since the machine is busy from time $\tau^*$ to $\tau_0$, the KRT assumption implies that no jobs arrive in the interval $(\tau^*, \tau_0)$. From Observation 1($a$), we have $r(B_0) = \tau_0$ and $r(\mathcal{Q}) = \tau_0$. By (8) and (5), we have that $C_{\text{opt}} \geqslant \tau_0 + p(B_0) + P(\mathcal{Q}) + P(\mathcal{R}) = C_{\text{on}}$. This contradicts (2) again. The assertion follows.

Suppose to the contrary that $r(\mathcal{Q}) > \tau^*$. Given that no jobs arrive in the interval $(\tau^*, \tau_0)$ and $r(\mathcal{Q}) \leqslant \tau_0$, we obtain $r(\mathcal{Q}) = \tau_0$. Recall that $r(\mathcal{R}) = \tau$ and $V(\tau) = \mathcal{Q} \cup \mathcal{R}$. Then $r(V(\tau)) = \tau_0$. Then, by (4), we have $C_{\text{opt}} \geqslant \tau_0 + P(\mathcal{Q}) + P(\mathcal{R})$. Furthermore, by (5), we obtain $C_{\text{on}} - C_{\text{opt}} < p(B_0)$. From the above assertion, we have $r(B_0) \leqslant \tau^*$. Note that $B^*$ is the longest waiting BCLPT batch in $\mathcal{A}(\tau^*)$ and $J(B_0) \notin B^*$. Then, by the BCLPT rule, $J(B_0)$ and $J(B^*)$ are incompatible and belong to two distinct BCLPT batches in $\mathcal{A}(\tau^*)$, and so $P(V(\tau^*)) \geqslant p(B^*) + p(B_0) \geqslant 2p(B_0)$. Using inequality (10), we can deduce that $C_{\text{opt}} \geqslant 2(1+\beta)p(B_0)$. We consider two cases as follows.

**Case 1** $0 < a \leqslant 1$. Then $(1+a)\beta \leqslant 2\beta$ and $2a \leqslant 2$, and so we have

$$\frac{C_{\text{on}} - C_{\text{opt}}}{C_{\text{opt}}} \leqslant \frac{p(B_0)}{2(1+\beta)p(B_0)} = \frac{1}{2\beta + 2} \leqslant \frac{1}{(1+a)\beta + 2a} = \beta.$$

**Case 2** $a > 1$. Then $1 + a < 2a$. From (6) and (7), we have $C_{\text{on}} - C_{\text{opt}} < \frac{1}{a}p(B_0)$, and so

$$\frac{C_{\text{on}} - C_{\text{opt}}}{C_{\text{opt}}} \leqslant \frac{\frac{1}{a}p(B_0)}{2(1+\beta)p(B_0)} = \frac{1}{2a \cdot \beta + 2a} < \frac{1}{(1+a)\beta + 2a} = \beta.$$

In conclusion, we deduce that $C_{\text{on}}/C_{\text{opt}} \leqslant 1 + \beta$, which contradicts (2) again. This completes our argument.

Claim 6 implies that at least one job from $\mathcal{Q}$ arrives at or before time $\tau^*$. Given that no jobs arrive in the interval $(\tau^*, \tau_0)$ and $r(\mathcal{Q}) \leqslant \tau_0$, then we can further divide $\mathcal{Q}$ into two subsets $\mathcal{Q}_1$ and $\mathcal{Q}_2$ such that

$$\mathcal{Q}_1 = \{J(B) : r(B) \leqslant \tau^*, J(B) \in \mathcal{Q}\} \text{ and } \mathcal{Q}_2 = \{J(B) : r(B) = \tau_0, J(B) \in \mathcal{Q}\}.$$

Then $P(\mathcal{Q}) = P(\mathcal{Q}_1) + P(\mathcal{Q}_2)$, and by (9), we have

$$C_{\text{on}} = \tau^* + p(B^*) + p(B_0) + P(\mathcal{Q}_1) + P(\mathcal{Q}_2) + P(\mathcal{R}). \tag{11}$$

Suppose that $\mathcal{Q}_1 = \{J(B_{i_{11}}), J(B_{i_{12}}), \cdots, J(B_{i_{1l}})\}$ and $p(B_{i_{1(j+1)}}) < p(B_{i_{1j}})$ for $1 \leqslant j \leqslant l-1$. Then we have following Observation 2 (similarly to Observation1).

**Observation 2** *(a)* $(\mathcal{Q}_1 \cup J(B^*)) \subseteq U(\tau^*)$; *(b) Every two jobs in* $\mathcal{Q}_1 \cup J(B^*)$ *are incompatible; furthermore,* $(1+a)p(B_{i_{1(j+1)}}) < p(B_{i_{1j}})$ *for* $1 \leqslant j \leqslant l-1$, *and* $(1+a)p(B_{i_{11}}) < p(B^*)$; *(c)* $P(V(\tau^*)) \geqslant p(B^*) + P(\mathcal{Q}_1)$.

From Observation 2*(b)*, we can deduce that

$$P(\mathcal{Q}_1) = \sum_{j=1}^{l} p(B_{i_{1j}}) < \left(\frac{1}{1+a} + \frac{1}{(1+a)^2} + \cdots + \frac{1}{(1+a)^l}\right) p(B^*) < \frac{1}{a} p(B^*). \tag{12}$$

**Claim 7** $r(B_0) \leqslant \tau^*$.

***Proof*** Suppose to the contrary that $r(B_0) > \tau^*$. Given that no jobs arrive in the interval $(\tau^*, \tau_0)$ and $J(B_0) \subseteq U(\tau_0)$, then $r(B_0) = \tau_0$. Note that $r(\mathcal{Q}_2) = \tau_0$, $r(\mathcal{R}) = \tau$ and $\mathcal{Q} = \mathcal{Q}_1 \cup \mathcal{Q}_2$. From Claim 4, we can deduce that $C_{\text{opt}} \geqslant \tau_0 + p(B_0) + P(\mathcal{Q}_2) + P(\mathcal{R})$. Combining (5), we have $C_{\text{on}} - C_{\text{opt}} \leqslant P(\mathcal{Q}) - P(\mathcal{Q}_2) = P(\mathcal{Q}_1)$. Write $\delta_1 = \min\{p(B_0), p(B^*)\}$. From (7) and (12), we have $P(\mathcal{Q}_1) < \frac{1}{a}\min\{p(B_0), p(B^*)\} = \frac{1}{a}\delta_1$. From Observation 2*(c)* and inequality (10), we have that $C_{\text{opt}} \geqslant \beta \cdot (p(B^*) + P(\mathcal{Q}_1)) + p(B^*) + p(B_0)$, and so

$$\begin{aligned} \frac{C_{\text{on}} - C_{\text{opt}}}{C_{\text{opt}}} &\leqslant \frac{P(\mathcal{Q}_1)}{\beta(p(B^*) + P(\mathcal{Q}_1)) + p(B^*) + p(B_0)} < \frac{\frac{1}{a}\delta_1}{\beta(\delta_1 + \frac{1}{a}\delta_1) + 2\delta_1} \\ &= \frac{1}{(1+a)\beta + 2a} = \beta. \end{aligned}$$

The last equality follows from (1). This contradicts (2) again. The claim follows.

Claim 7 implies that $J(B_0) \subseteq U(\tau^*)$. Note that $\mathcal{Q}_1 \subseteq U(\tau^*)$, $J(B^*) \subseteq U(\tau^*)$ and $(\mathcal{Q}_1 \cup J(B_0)) \cap B^* = \varnothing$. Then, by the BCLPT rule, any job in $\mathcal{Q}_1 \cup J(B_0)$ is incompatible with $J(B^*)$. Note that $B^*$ is the longest waiting BCLPT batch at time $\tau^*$. From Observation 2 and Observation 1($b$), we can further get following Observation 3.

**Observation 3** *($a$) $(\mathcal{Q}_1 \cup J(B_0) \cup J(B^*)) \subseteq U(\tau^*)$; ($b$) every two jobs in $\mathcal{Q}_1 \cup J(B_0) \cup J(B^*)$ are incompatible; furthermore, $(1+a)p(B_{i_{1(j+1)}}) < p(B_{i_{1j}})$ for $1 \leqslant j \leqslant l-1$, $(1+a)p(B_{i_{11}}) \leqslant (1+a)p(B_{i_1}) < p(B_0)$ and $(1+a)p(B_0) < p(B^*)$; ($c$) $P(V(\tau^*)) \geqslant p(B^*) + p(B_0) + P(\mathcal{Q}_1)$.*

From Observation 3($b$), we can deduce that

$$p(B_0) + P(\mathcal{Q}_1) < \left(\frac{1}{1+a} + \frac{1}{(1+a)^2} + \cdots + \frac{1}{(1+a)^{l+1}}\right) p(B^*) < \frac{1}{a} p(B^*). \tag{13}$$

By Observation 3($c$) and inequality (10), we have that

$$C_{\mathrm{opt}} \geqslant \beta \cdot (p(B^*) + p(B_0) + P(\mathcal{Q}_1)) + p(B^*) + p(B_0). \tag{14}$$

Recall that every two jobs from $\mathcal{Q} \cup \mathcal{R}$ are incompatible. Note that $r(\mathcal{Q}_2) = \tau_0$, $r(\mathcal{R}) = \tau$ and $\mathcal{Q}_2 \subseteq \mathcal{Q}$. Then we have $C_{\mathrm{opt}} \geqslant \tau_0 + P(\mathcal{Q}_2) + P(\mathcal{R})$. And so, by (5), we obtain $C_{\mathrm{on}} - C_{\mathrm{opt}} \leqslant p(B_0) + P(\mathcal{Q}_1)$. Write $\delta_2 = \min\{P(\mathcal{Q}), p(B_0) + P(\mathcal{Q}_1)\}$. Then, combining (6), we obtain $C_{\mathrm{on}} - C_{\mathrm{opt}} \leqslant \delta_2$. Note that $p(B_0) < p(B^*)$. By (7) and (13), we have $\delta_2 < \frac{1}{a} p(B_0)$. From (14), we have $C_{\mathrm{opt}} > \beta\delta_2 + (2+\beta)p(B_0)$, and so

$$\frac{C_{\mathrm{on}} - C_{\mathrm{opt}}}{C_{\mathrm{opt}}} < \frac{\delta_2}{\beta\delta_2 + (2+\beta)p(B_0)} < \frac{\frac{1}{a}p(B_0)}{(\frac{\beta}{a} + 2 + \beta)p(B_0)} = \frac{1}{(1+a)\beta + 2a} = \beta.$$

This contradicts (2) again.

The above discussions imply that $\mathcal{I}$ is not a counter-example. Then following Theorem 1 holds:

**Theorem 1** *For problem $1|online, KRT, p\text{-}batch, G = a\text{-}INT|C_{\max}$, the competitive ratio of algorithm $H$ is not greater than $1+\beta$, where $\beta = \sqrt{\lambda^2 - \lambda + 1} - \lambda$ and $\lambda = \frac{a}{1+a}$.*

## 4 A Matching Lower Bound

In this section we prove that, because of the lack of information concerning the future, there cannot exist any online algorithm with competitive ratio smaller than $1+\beta$. Recall that $\beta = \sqrt{\lambda^2 - \lambda + 1} - \lambda$ satisfies the equation $x^2 + 2\lambda x + (\lambda - 1) = 0$, where $\lambda = \frac{a}{1+a}$. Since $a > 0$, then $0 < \lambda < 1$ and $0 < \beta < 1$.

**Theorem 2** *For problem $1|online, KRT, p\text{-}batch, G = a\text{-}INT|C_{\max}$, there does not exist any online algorithm with a competitive ratio strictly smaller than $1+\beta$.*

***Proof*** To prove the lower bound $1+\beta$, we introduce the following instance presented by Fu et al. [17] for problem $1|\text{online}, r_j, \text{p-batch}, \text{G} = a\text{-INT}|C_{\max}$. However, in order to satisfy the KRT assumption, we appropriately revise this instance. The proof is based on an adversary argument.

The initial instance consists of $k$ jobs $J_1, J_2, \cdots, J_k$ that arrive at time 0 with normal processing times

$$p_1 = 1,\ p_2 = \tfrac{1}{1+a}(p_1 - \varepsilon),\ p_3 = \tfrac{1}{1+a}(p_2 - \varepsilon),\ \cdots,\ p_k = \tfrac{1}{1+a}(p_{k-1} - \varepsilon),$$

where $k$ is a sufficiently large positive integer and $\varepsilon = \frac{a}{k!}$. We can readily observe that $0 < p_i < \frac{1}{1+a}p_{i-1}$ for $2 \leqslant i \leqslant k$. Then, by the definition of job compatibility, we know that any pair of these $k$ jobs are incompatible, and they will independently form a batch processed in any schedule. Let $P = \sum_{j=1}^{k} p_j$ and $\beta_k = \frac{\sqrt{P^2-P+1}-1}{P}$ (satisfying the equation $Px^2+2x = P-1$). If an online algorithm starts processing a job $J_i$ with $1 \leqslant i \leqslant k$ in time interval $[0, \beta_k P)$, then the adversary releases a new copy of $J_i$, i.e., a job with the same normal processing time as $J_i$, at the completion time of job $J_i$ (satisfying the KRT assumption). If one job's starting time is equal to or greater than $\beta_k P$, no jobs will come and the job sequence terminates. We use $\mathcal{A}_i$ to denote the set containing job $J_i$ and its replicas (the number of jobs in $\mathcal{A}_i$ may be larger than two). Suppose that the jobs with the starting times in time interval $[0, \beta_k P)$ belong to $h$ job sets $\mathcal{A}_{i_1}, \mathcal{A}_{i_2}, \ldots, \mathcal{A}_{i_h}$ (if there does not exist any job with the starting time in the interval $[0, \beta_k P)$, we define $h = 0$), and the remaining $(k-h)$ job sets are denoted by $\mathcal{A}_{i_{h+1}}, \mathcal{A}_{i_{h+2}}, \cdots, \mathcal{A}_{i_k}$. For any $j \in \{1, 2 \cdots, h\}$, let $S_j$ be the largest starting time of all jobs in $\mathcal{A}_{i_j}$ in the time interval $[0, \beta_k P)$. Then, according to the adversary's strategy, we know that the maximum release time of the jobs in $\mathcal{A}_{i_j}$ with $1 \leqslant j \leqslant h$ is equal to $S_j + p_{i_j}$. Furthermore, we may suppose that $S_1 < S_2 < \cdots < S_h$. Then we have

$$S_{j+1} \geqslant S_j + p_{i_j} \quad \text{for} \quad 1 \leqslant j \leqslant h-1. \tag{15}$$

Given that $0 < \beta_k < 1$ and $\sum_{i=1}^{k} p_i = P$, there does exist at least one job from $\mathcal{A}_{i_j}$ with $1 \leqslant j \leqslant k$ which has a completion time not smaller than $\beta_k P$. Let $J_e$ be a such job with the minimum starting time, and its starting time is denoted by $S$. Then $S \geqslant \beta_k P - p_e$. We will distinguish two cases depending on the value of $S$.

**Case 1** $S \geqslant \beta_k P$. In this case, we can observe that there still exist $k$ mutually incompatible jobs unprocessed at time $S$ (the case $h = 0$ is included). Then $C_{\text{on}} \geqslant S + \sum_{i=1}^{k} p_i = S + P \geqslant (1+\beta_k)P$. We consider two subcases as follows.

**Subcase 1.1** $h = 0$. In this case, no job has the starting time smaller than $\beta_k P$, and the job instance consists of the first $k$ jobs. Recall that these $k$ jobs arrive at time 0 and any pair of them are incompatible. Then it can be easily obtained that $C_{\text{opt}} = P$.

**Subcase 1.2** $h \neq 0$. In this case, by the definitions of $h$ job sets $\mathcal{A}_{i_1}, \mathcal{A}_{i_2}, \cdots, \mathcal{A}_{i_h}$, we know that $S_h + p_{i_h} < \beta_k P = \sqrt{P^2 - P + 1} - 1 < P - 1$; the last inequality follows from $P > 1$. Recall that the maximum release time of the jobs in $\mathcal{A}_{i_j}$ with $1 \leqslant j \leqslant h$ is equal to $S_j + p_{i_j}$ (strictly smaller than $P-1$), and the only one job in $\mathcal{A}_{i_j}$ with $h+1 \leqslant j \leqslant k$ arrives at time 0. Since inequality (15) holds and

$\max\limits_{1\leqslant l\leqslant k}\{p_{i_l}\} = \max\limits_{1\leqslant i\leqslant k}\{p_k\} = 1$, we can obtain an optimal schedule $\pi$: $\forall j \in \{1, \cdots, k\}$, all jobs in $\mathcal{A}_{i_j}$ form a single batch and start at time $S_j(\pi)$, where

$$\begin{cases} S_j(\pi) = 0, & \text{if } j = k, \\ S_j(\pi) = \sum_{l=j+1}^{k} p_{i_l}, & \text{if } h+1 \leqslant j \leqslant k-1, \\ S_j(\pi) = P - \sum_{l=j}^{h} p_{i_l}, & \text{if } 1 \leqslant j \leqslant h. \end{cases}$$

Note that $\sum_{l=h+1}^{k} p_{i_l} + \sum_{l=1}^{h} p_{i_l} = \sum_{l=1}^{k} p_{i_l} = \sum_{j=1}^{k} p_j = P$. Then $C_{\text{opt}} = \sum_{l=1}^{k} p_{i_l} = P$.

In both subcases, we get $C_{\text{opt}} = P$, which, together the fact that $C_{\text{on}} \geqslant (1+\beta_k)P$ in Case 1, leads to $C_{\text{on}}/C_{\text{opt}} \geqslant 1 + \beta_k$.

**Case 2** $S \in [\beta_k P - p_e, \beta_k P)$. In this case, we can find that $J_e \in \mathcal{A}_{i_h}$, $p_e = p_{i_h}$, $S_h = S$, and there are still $k$ unprocessed jobs at time $S + p_e$ such that they are mutually incompatible. Then we have $C_{\text{on}} \geqslant S + p_e + \sum_{i=1}^{k} p_i = S + p_e + P$. We consider two subcases as follows.

**Subcase 2.1** $h = k$. In this case, for each $j \in \{1, 2, \cdots, k\}$, the maximum release time of the jobs in $\mathcal{A}_{i_j}$ is $S_j + p_{i_j}$ (not larger than $S_j + 1$). Given that inequality (15) holds, a feasible schedule $\pi'$ can be established in the following way: $\forall j \in \{1, \cdots, k\}$, all jobs $\mathcal{A}_{i_j}$ form a single batch and start at $S_j(\pi') = S_j + 1$. Then we have $C_{\text{opt}} \leqslant C_{\max}(\pi') = S_h(\pi') + p_{i_h} = S_h + 1 + p_{i_h} = S + 1 + p_e$.

**Subcase 2.2** $h < k$. Note that the release time of the only one job in $\mathcal{A}_{i_j}$ with $h+1 \leqslant j \leqslant k$ is 0, and the maximum release time of the jobs in $\mathcal{A}_{i_j}$ with $1 \leqslant j \leqslant h$ is $S_j + p_{i_j}$ (not larger than $S_j + 1$). Given that inequality (15) holds, a feasible schedule $\pi'$ can be constructed as follows: $\forall j \in \{1, \cdots, k\}$, all jobs $\mathcal{A}_{i_j}$ form a single batch and start at $S_j(\pi')$, where

$$\begin{cases} S_j(\pi') = 0, & \text{if } j = k, \\ S_j(\pi') = \sum_{l=j+1}^{k} p_{i_l}, & \text{if } h+1 \leqslant j \leqslant k-1, \\ S_j(\pi') = \max\{S_j + 1, \sum_{l=h+1}^{k} p_{i_l} + \sum_{l=1}^{j-1} p_{i_l}\}, & \text{if } 1 \leqslant j \leqslant h. \end{cases}$$

Then we have

$$C_{\max}(\pi') = S_h(\pi') + p_{i_h} = \max\{S_h + 1, \sum_{l=h+1}^{k} p_{i_l} + \sum_{l=1}^{h-1} p_{i_l}\} + p_{i_h}.$$

Recall that $S_h = S$, $p_{i_h} = p_e$ and $\sum_{l=1}^{k} p_{i_l} = \sum_{j=1}^{k} p_j = P$. Therefore, we have $C_{\text{opt}} \leqslant C_{\max}(\pi') = \max\{S + 1 + p_e,\ P\}$.

In both subcases, we can deduce $C_{\text{opt}} \leqslant \max\{S + 1 + p_e,\ P\}$. Recall that $C_{\text{on}} \geqslant S + p_e + P$. If $C_{\text{opt}} \leqslant S + 1 + p_e$, noting that $p_e \leqslant \max\limits_{1\leqslant i\leqslant k}\{p_k\} = 1$ and $S < \beta_k P$, then we obtain

$$\frac{C_{\text{on}}}{C_{\text{opt}}} \geqslant \frac{S+p_e+P}{S+1+p_e} = 1+\frac{P-1}{S+1+p_e} > 1+\frac{P-1}{\beta_k P+1+1} = 1+\beta_k,$$

where the last equality follows from that $\beta_k$ satisfies the equation $Px^2+2x = P-1$. If $C_{\text{opt}} \leqslant P$, noting that $S \geqslant \beta_k P - p_e$, we have

$$\frac{C_{\text{on}}}{C_{\text{opt}}} \geqslant \frac{S+p_e+P}{P} \geqslant \frac{\beta_k P - p_e + p_e + P}{P} = 1+\beta_k.$$

In conclusion, we can finally obtain $C_{\text{on}}/C_{\text{opt}} \geqslant 1+\beta_k$. Note that $P = \sum_{j=1}^{k} p_j = \sum_{i=0}^{k-1} \frac{1}{(1+a)^i} - \sum_{i=1}^{k} \frac{k-i}{(1+a)^i}\varepsilon$. Then $\lim\limits_{k\to\infty} P = \frac{1+a}{a} = \frac{1}{\lambda}$, and so $\beta_k = \frac{\sqrt{P^2-P+1}-1}{P}$ tends to $\beta = \sqrt{\lambda^2-\lambda+1}-\lambda$ as $k$ tends to $\infty$. Thus we get

$$\frac{C_{\text{on}}}{C_{\text{opt}}} \geqslant 1+\beta_k \to 1+\beta, \text{ as } k \to \infty.$$

This completes the proof.

Combining Theorem 1 with Theorem 2, we have established the following theorem.

**Theorem 3** *For problem* $1|online, KRT, p\text{-}batch, G = a\text{-}INT|C_{\max}$*, algorithm* $H$ *is a best possible online algorithm with a competitive ratio of* $1+\beta$*, where* $\beta = \sqrt{\lambda^2-\lambda+1}-\lambda$ *and* $\lambda = \frac{a}{1+a}$.

## Conclusions

In this paper, we study the problem of the online scheduling with kind release times and job compatibilities on a single unbounded parallel-batch machine to minimize makespan. We present a best possible online algorithm with competitive ratio of $1+\sqrt{\lambda^2-\lambda+1}-\lambda$, where $\lambda = \frac{a}{1+a}$.

**Acknowledgements** The authors would like to thank the editor and two anonymous referees for their helpful suggestions which have significantly improved the presentation of this paper.

**Author Contributions** All authors contributed equally to this paper.

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

## References

[1] Lee, C.Y., Uzsoy, R., Martin-Vega, L.A.: Efficient algorithms for scheduling semiconductor burn-in operations. Oper. Res. **40**, 764–775 (1992)

[2] Brucker, P., Gladky, A., Hoogveen, H., Kovalyov, M.Y., Potts, C.N., Tautenhahn, T., van de Velde, S.L.: Scheduling a batching machine. J. Sched. **1**, 31–54 (1998)

[3] Zhang, G.C., Cai, X.Q., Wong, C.K.: On-line algorithms for minimizing makespan on batch processing machines. Nav. Res. Log. **48**, 241–258 (2001)

[4] Dobson, G., Nambimadom, R.S.: The batch loading and scheduling problem. Oper. Res. **49**, 52–65 (2001)

[5] Deng, X.T., Poon, C.K., Zhang, Y.Z.: Approximation algorithms in batch processing. J. Comb. Optim. **7**, 247–257 (2003)

[6] Poon, C.K., Yu, W.C.: A flexible on-line scheduling algorithm for batch machine with infinite capacity. Ann. Oper. Res. **133**, 175–181 (2005)

[7] Liu, P.H., Lu, X.W., Fang, Y.: A best possible deterministic on-line algorithm for minimizing makespan on parallel batch machines. J. Sched. **15**, 77–81 (2012)

[8] Tian, J., Fu, R.Y., Yuan, J.J.: Online over time scheduling on parallel-batch machines: a survey. J. Oper. Res. Soc. China **2**, 445–454 (2014)

[9] Li, W.H., Chai, X.: Online scheduling on bounded batch machines to minimize the maximum weighted completion time. J. Oper. Res. Soc. China **6**, 455–465 (2018)

[10] Yuan, J.J., Ren, L.L., Tian, J., Fu, R.Y.: Online scheduling on two uniform unbounded parallel-batch machines to minimize makespan. J. Oper. Res. Soc. China **7**, 303–319 (2019)

[11] Liu, H.L., Lu, X.W.: Online scheduling on a parallel batch machine with delivery times and limited restarts. J. Oper. Res. Soc. China **10**, 113–131 (2022)

[12] Bellanger, A., Janiak, A., Kovalyov, M.Y., Oulamara, A.: Scheduling an unbounded batching machine with job processing time compatibilities. Discrete Appl. Math. **160**, 15–23 (2012)

[13] Boudhar, M., Finke, G.: Scheduling on a batch machine with job compatibilities. Belg. J. Oper. Res. Stat. Comput. Sci. **40**, 69–80 (2000)

[14] Boudhar, M.: Scheduling a batch processing machine with bipartite compatibility graph. Math. Method. Oper. Res. **57**, 327–513 (2003)

[15] Finke, G., Jost, V., Queyranne, M., Sebo, A.: Batch processing with interval compatibilities between tasks. Discrete Appl. Math. **156**, 556–568 (2008)

[16] Li, S.S., Cheng, T.C.E., Ng, C.T., Yuan, J.J.: Single-machine batch scheduling with job processing time compatibilities. Theor. Comput. Sci. **583**, 57–66 (2015)

[17] Fu, R.Y., Tian, J., Yuan, J.J., Li, S.S.: An optimal online algorithm for the parallel-batch scheduling with job processing time compatibilities. J. Comb. Optim. **34**, 1187–1197 (2017)

[18] Li, W.J., Yuan, J.J.: LPT online strategy for parallel-machine scheduling with kind release times. Optim. Lett. **10**, 159–168 (2016)

[19] Li, W.J., Li, S.S., Feng, Q.: Online batch scheduling with kind release times and incompatible families to minimize makespan. Optim. Lett. **12**, 301–310 (2018)