J Comb Optim (2015) 29:228–236
DOI 10.1007/s10878-014-9706-4

Online unbounded batch scheduling on parallel machines with delivery times

Peihai Liu · Xiwen Lu

Published online: 30 January 2014

Abstract We consider the online unbounded batch scheduling problems on m identical machines subject to release dates and delivery times. Jobs arrive over time and the characteristics of jobs are unknown until their arrival times. Jobs can be processed in a common batch and the batch capacity is unbounded. Once the processing of a job is completed it is independently delivered to the destination. The objective is to minimize the time by which all jobs have been delivered. For each job J_j, its processing time and delivery time are denoted by p_j and q_j , respectively. We first consider a restricted model: the jobs have agreeable processing and delivery times, i.e., for any two jobs J_i and J_j $p_i > p_j$ implies $q_i \geq q_j$. For the restrict case, we provide a best possible online algorithm with competitive ratio $1 + \alpha_m$, where $\alpha_m > 0$ is determined by $\alpha_m^2 + m\alpha_m = 1$. Then we present an online algorithm with a competitive ratio of $1 + 2/\lfloor\sqrt{m}\rfloor$ for the general case.

Keywords Scheduling · Online algorithm · Batch machines · Delivery times · Competitive ratio

1 Introduction

We consider an online scheduling model: online unbounded batch scheduling on parallel machines with delivery times. Here, we have m batch machines and sufficiently many vehicles. There are n jobs $J_1, J_2, \ldots, J_n$. Each job has a release date r_j, a processing time p_j, and a delivery time q_j. These characteristics about a job are unknown until it arrives. In this model, one batch machine can handle up to B jobs

P. Liu (✉)· X. Lu
Department of Mathematics, East China University of Science and Technology,
Shanghai 200237, China
e-mail: pliu@ecust.edu.cn

simultaneously as a batch, where B is sufficiently large. The processing time for a batch is equal to the longest processing time in the batch. All jobs in a common batch have the same starting time and completion time. Each job needs to be processed on one of the m batch machines, and once the job is completed we deliver it immediately to the destination by a vehicle. The objective is to minimize the time by which all jobs have been delivered. We denote by S_j, C_j and L_j, respectively, the starting time, the completion time and the time by which J_j is delivered in a schedule. By using the general notation for a schedule problem, introduced by Graham et al. (1979), this problem is denoted by $Pm|online, r_j, q_j, B = \infty|L_{max}$, where $L_{max} = \max\{L_j : L_j = C_j + q_j, 1 \le j \le n\}$.

Batch scheduling was first introduced by Lee et al. (1992), which was motivated by burn-in operations in semiconductor manufacturing. Deng et al. (2003) and Zhang et al. (2001) studied the online scheduling problem on single batch machine. They proved that there is no online algorithm with competitive ratio smaller than $(\sqrt{5}+1)/2$, and for the case $B = \infty$, they independently gave the same online algorithm with competitive ratio matching the lower bound. For the case $B < n$, the first online algorithm is a greedy heuristic, GRLPT, of Lee and Uzsoy (1999) which was shown to be 2-competitive by Liu and Yu (2000). Zhang et al. (2001) presented two online algorithms with competitive ratio not greater than 2. The above three online algorithms are all based on the ideas of the FBLPT rule. Later, Poon and Yu (2005) presented a class of algorithms called the FBLPT-based algorithms that contains all the above three algorithms as special cases, and showed that any FBLPT-based algorithm has competitive ratio at most 2. In particular, for the case B = 2, they also gave an online algorithm with competitive ratio 7/4.

For the online model of scheduling on m unbounded parallel batch machines, Zhang et al. (2001) gave a lower bound $\sqrt[m]{2}$, and presented an online algorithm with competitive ratio $1 + \alpha_m$, where $\alpha_m = (1 - \alpha_m)^{m-1}$. For the special case where $m = 2$, Nong et al. (2008) gave a $\sqrt{2}$-competitive online algorithm. For the general case, Liu et al. (2012) and Tian et al. (2009) showed the lower bound is $1 + (\sqrt{m^2+4} - m)/2$, and gave different optimal algorithms independently.

There have also been some results about online scheduling problems with delivery times. Hoogeveen and Vestjean (2000) first studied the single-machine online scheduling problem with delivery times. They showed the lower bound is $(\sqrt{5}+1)/2$, and gave a best possible deterministic online algorithm. On identical parallel machines, Vestjens (1997) showed that no online algorithms can have competitive ratio smaller than 1.5 and showed the competitive ratio of the LS algorithm is 2. Tian et al. (2007) considered the single batch machine online scheduling problem with delivery times. They gave a 2-competitive algorithm for the unbounded case and a 3-competitive algorithm for the bounded case. In the same paper, they also studied a special model with identical processing times. They provided the best online algorithms with competitive ratio $(\sqrt{5}+1)/2$ for both bounded and unbounded cases. In another paper, Tian et al. (2012) gave an improved online algorithm for the same problem with competitive ratio $2\sqrt{2} - 1$. Yuan et al. (2009) also studied the single machine parallel-batch scheduling. They provided a best possible online algorithm for two restricted models. Fang et al. (2011) addressed online scheduling on m unbounded batch machines with delivery times. They gave an online algorithm with competitive ratio $1.5 + o(1)$.

In this paper, we investigate online algorithms for parallel machine batch scheduling with delivery times. We first consider a restricted model: the jobs have agreeable processing and delivery times, i.e., for any two jobs J_i and J_j, $p_i > p_j$ implies $q_i \geq q_j$. We provide a best possible online algorithm with competitive ratio $1 + \alpha_m$ for the problem, where $\alpha_m^2 + m\alpha_m = 1$. We also study the general case. We provide an online algorithm with a competitive ratio of $r = 1 + 1/u + 1/v$, where u, v are integers and $uv \leq m$. Especially if we take $u = v = \lfloor\sqrt{m}\rfloor$ and let $m \to +\infty$, then $r \to 1$. This improves the current algorithm with competitive ratio a 1.5+o(1) for this problem in the literature.

Throughout this paper, we use $r(S)$, $p(S)$, $q(S)$ to denote the smallest release date, the largest processing time and the largest delivery time of jobs in S, respectively.

2 A lower bound

We first consider the scheduling model $Pm|online, r_j, B = \infty|C_{max}$, which is a special case of the scheduling problems studied in this paper. Hence, its lower bound is also a lower bound of the problems we study. For the former, Liu et al. (2012) and Tian et al. (2009) presented the following lower bound of competitive ratio for all online algorithms independently.

Lemma 1 [Liu et al. (2012), Tian et al. (2009)] There is no online algorithm with competitive ratio less than $1 + \alpha_m$ for $Pm|online, r_j, B = \infty|C_{max}$, where $\alpha_m^2 + m\alpha_m = 1$.

Corollary 1 *There is no on-line algorithm with competitive ratio less than* $1 + \alpha_m$ *for the following two problems where* $\alpha_m^2 + m\alpha_m = 1$,

(1) $Pm|online, r_j, agreeable(p_j, q_j), B = \infty|L_{max}$
(2) $Pm|online, r_j, B = \infty|L_{max}$

3 A restrict case

In this section, we deal with a restrict model: jobs have agreeable processing and delivery times, i.e., for any two jobs J_i and J_j $p_i > p_j$ implies $q_i \geq q_j$. We provide a best possible online algorithm with competitive ratio $1 + \alpha_m$ for the problem, where $\alpha_m = (\sqrt{m^2 + 4} - m)/2$, i.e. $\alpha_m^2 + m\alpha_m = 1$.

Let $A(t)$ be the set of the jobs which are available but not yet scheduled at time t. Denote by $p(t)$ the largest processing time of jobs in $A(t)$. Denote by $r(t)$ the smallest release date of the jobs in $A(t)$.

Algorithm $H1$ Whenever a machine is idle and $A(t) \neq \emptyset$, make decision as following: If $t \geq (1 + \alpha_m)r(t) + \alpha_m p(t)$, then start all the available jobs as a single batch on the idle machine. Otherwise, do nothing but wait.

Now, we will prove that $H1$ has a competitive ratio of $1 + \alpha_m$. Let σ and π denote the schedule generated by algorithm $H1$ and an optimal off-line schedule, respectively. Their objective values are denoted by $L_{max}(\sigma)$ and $L_{max}(\pi)$, respectively. We assume that there are b batches totally in σ which are written as $B_1, B_2, \ldots, B_b$. For each i, the longest job in batch B_i is denoted by J_i(if two or more jobs have the longest processing time in batch B_i, let J_i be the job with a larger delivery time) with a release date r_i, a processing time p_i and a delivery time q_i. Since the jobs have agreeable processing and delivery times, $p_i = p(B_i), q_i = q(B_i)$. Thus we can use $S_i(\sigma)$ and $C_i(\sigma)$ to denote the starting time and the completion time of both job J_i and batch B_i in σ, respectively. It can be observed that $S_i(\sigma) < S_j(\sigma)$ implies $r(B_j) > S_i(\sigma)$ and $S_j(\sigma) \geq (1+\alpha_m)r(B_j) + \alpha_m p_j > (1+\alpha_m)S_i(\sigma) + \alpha_m p_j$. For any batch B_i, if $S_i(\sigma) = (1+\alpha_m)r(B_i) + \alpha_m p(B_i)$, we say that B_i is regular. For convenience, we assume that $S_1(\sigma) < S_2(\sigma) < \cdots < S_b(\sigma)$.

Property 1 *For any two batches in σ, say B_i and B_j, if $S_i(\sigma) < S_j(\sigma)$, then $r_j > S_i(\sigma)$ and $S_j(\sigma) \geq (1+\alpha_m)r_j > (1+\alpha_m)S_i(\sigma)$.*

Lemma 2 *In σ, if B_k is not regular, then $S_{k-1}(\sigma) > (1-\alpha_m)S_k(\sigma)$.*

Proof B_k is not regular means that $S_k(\sigma) > (1+\alpha_m)r(B_k) + \alpha_m p_k$. Then by the algorithm, each machine is busy processing a batch during $[r_k, S_k(\sigma))$. Otherwise, B_k will start earlier in σ. Denote the m batches processed during $[r_k, S_k(\sigma))$ by $B_{k_1}, B_{k_2}, \ldots, B_{k_m}$ with $S_{k_1}(\sigma) < S_{k_2}(\sigma) < \cdots < S_{k_m}(\sigma)$. It is clear that $k_m = k-1$ and $C_{k_j}(\sigma) = S_{k_j}(\sigma) + p_{k_j} \geq S_k(\sigma)(j = 1, 2, \ldots, m)$, i.e.,

$$p_{k_j} \geq S_k(\sigma) - S_{k_j}(\sigma),\ j = 1, 2, \ldots, m \tag{1}$$

By the algorithm, $S_{k_1}(\sigma) \geq \alpha_m p_{k_1} \geq \alpha_m(S_k(\sigma) - S_{k_1}(\sigma))$. Thus

$$S_{k_1}(\sigma) \geq \frac{\alpha_m}{1+\alpha_m} S_k(\sigma) \tag{2}$$

In addition, for each $j(2 \leq j \leq m)$, $S_{k_j}(\sigma) \geq (1+\alpha_m)r(B_{k_j}) + \alpha_m p(B_{k_j}) > (1+\alpha_m)S_{k_{j-1}}(\sigma) + \alpha_m p_{k_j}$. Using the inequality (1), we deduce that $S_{k_j}(\sigma) \geq (1+\alpha_m)S_{k_{j-1}}(\sigma) + \alpha_m(S_k(\sigma) - S_{k_j}(\sigma))$, i.e.,

$$S_{k_j}(\sigma) > S_{k_{j-1}}(\sigma) + \frac{\alpha_m}{1+\alpha_m} S_k(\sigma),\ 2 \leq j \leq m \tag{3}$$

Combining (2) and (3) we have that

$$S_{k_m}(\sigma) > \frac{m\alpha_m}{1+\alpha_m} S_k(\sigma) = (1-\alpha_m)S_k(\sigma) \tag{4}$$

i.e. $S_{k-1}(\sigma) > (1-\alpha_m)S_k(\sigma)$. □

Lemma 3 *If B_j is not regular, then $p_j < \alpha_m S_j(\sigma)$.*

Proof If B_j is not a regular batch, then from Lemma 2 we have that $S_{j-1}(\sigma) > (1-\alpha_m)S_j(\sigma)$. By the algorithm, we know that $S_j(\sigma) > (1+\alpha_m)S_{j-1}(\sigma) + \alpha_m p_j$. Thus, $S_j(\sigma) > (1-\alpha_m^2)S_j(\sigma) + \alpha_m p_j$ which implies that $p_j < \alpha_m S_j(\sigma)$.

This competes the proof. □

Lemma 4 *If B_k is not regular, then each machine is busy processing a regular batch during $[r_k, S_k(\sigma))$.*

Proof B_k is not regular means that $S_k(\sigma) > (1+\alpha_m)r(B_k) + \alpha_m p_k$. Then by the algorithm, each machine is busy processing a batch during $[r_k, S_k(\sigma))$. Denote the m batches processed in $[r_k, S_k(\sigma))$ by $B_{k_1}, B_{k_2}, \ldots, B_{k_m}$. Then

$$S_{k_j}(\sigma) + p_{k_j} \geq S_k(\sigma),\ j = 1, 2, \ldots, m \tag{5}$$

Suppose for the sake of contradiction that B_{k_j} is not regular for some j. Then by Lemma 3, $p_{k_j} < \alpha_m S_{k_j}$. Thus $S_k(\sigma) \geq (1+\alpha_m)r(B_k) > (1+\alpha_m)S_{k_j}(\sigma) > S_{k_j}(\sigma) + p_{k_j}$ which contradicts the inequality (5). Therefore, B_{k_j} is regular for each $j (1 \leq j \leq m)$.

This competes the proof. □

Theorem 1 $L_{max}(\sigma) \leq (1+\alpha_m)L_{max}(\pi)$.

Proof Let B_l denote the first batch in σ that assumes the objective value $L_{max}(\sigma)$, i.e.,

$$L_{max}(\sigma) = S_l(\sigma) + p_l + q_l \tag{6}$$

If B_l is regular, then $S_l(\sigma) = (1+\alpha_m)r(B_l) + \alpha_m p_l$. Thus $L_{max}(\sigma) = (1+\alpha_m)(r(B_l)+p_l)+q_l$. It is clear that $L_{max}(\pi) \geq r(B_l)+p_l+q_l$. Therefore, $L_{max}(\sigma) \leq (1+\alpha_m)L_{max}(\pi)$.

If B_l is not regular, then $S_l(\sigma) > (1+\alpha_m)r(B_l) + \alpha_m p_l$. Thus by the algorithm, each machine is busy processing a batch during $[r_l, S_l(\sigma))$. Denote the m batches processed in $[r_l, S_l(\sigma))$ by $B_{l_1}, B_{l_2}, \ldots, B_{l_m}$ with $S_{l_1}(\sigma) < S_{l_2}(\sigma) < \cdots < S_{l_m}(\sigma)$. It is clear that $l_m = l - 1$ and

$$S_l(\sigma) \leq S_{l_j}(\sigma) + p_{l_j},\ j = 1, 2, \ldots, m \tag{7}$$

According to Lemma 4, B_{l_j} is regular, i.e.

$$S_{l_j}(\sigma) = (1+\alpha_m)r(B_{l_j}) + \alpha_m p_{l_j} \leq (1+\alpha_m)r_{l_j} + \alpha_m p_{l_j} \tag{8}$$

Now, we will consider two cases according to the assignment of the m jobs $J_{l_j} (1 \leq j \leq m)$ in the optimal schedule π.

Case 1 $S_{l_j}(\pi) < S_{l_j}(\sigma)$ for each $j (1 \leq j \leq m)$. Then $r_{l_j} > S_{l_{j-1}}(\sigma) > S_{l_{j-1}}(\pi)$ for each $j (2 \leq j \leq m)$. Hence, the m jobs $J_{l_j} (1 \leq j \leq m)$ are processed in m different batches in π.

From the inequality (7), (8), we deduce that $S_l(\sigma) \leq S_{l_j}(\sigma) + p_{l_j} \leq (1+\alpha_m)(r_{l_j} + p_{l_j})$ for each $j = 1, 2, \ldots, m$. Therefore, $C_{l_j}(\pi) \geq r_{l_j} + p_{l_j} \geq S_l(\sigma)/(1+\alpha_m) > S_{l_m}(\sigma)$. Recall that $S_{l_j}(\pi) < S_{l_m}(\sigma)$ for each $j (1 \leq j \leq m)$. Hence the m jobs $J_{l_j} (1 \leq j \leq m)$ are grouped into m batches which are processed on m different machines in the optimal schedule π. Recall that $r_l > S_{l_m}(\sigma) \geq S_{l_j}(\pi)$. Thus, in the optimal schedule π, J_l starts after some job J_{l_j}. So

$$L_{max}(\pi) \geq \min_{1 \leq j \leq m} \left\{r_{l_j} + p_{l_j}\right\} + p_l + q_l \tag{9}$$

While $L_{max}(\sigma) \leq \min_j\{S_{l_j}(\sigma)+p_{l_j}\}+p_l+q_l \leq (1+\alpha_m)\min_j\{r_{l_j}+p_{l_j}\}+p_l+q_l$. Therefore, $L_{max}(\sigma) \leq (1+\alpha_m)L_{max}(\pi)$.
Case 2 $S_{l_j}(\pi) \geq S_{l_j}(\sigma)$ for some $j(1 \leq j \leq m)$. Then

$$L_{max}(\pi) \geq S_{l_j}(\sigma) + p_{l_j} \geq S_l(\sigma) \tag{10}$$

By Lemma 3, we have $S_{l_m}(\sigma) > (1-\alpha_m)S_l(\sigma)$. Thus

$$L_{max}(\pi) \geq r_l + p_l + q_l > S_{l_m}(\sigma) + p_l + q_l \geq (1-\alpha_m)S_l(\sigma) + p_l + q_l \tag{11}$$

Then comuting the inequality $\alpha_m\cdot$(10)+ (11) and using (6) we deduce that $L_{max}(\sigma) \leq (1+\alpha_m)L_{max}(\pi)$.
This completes the proof. □

4 The general case

In this section, we present an online algorithm for the general case which has a competitive ratio of $1 + 1/u + 1/v$, where u, v are positive integers and $uv \leq m$.

Select uv machines and partition the uv machines into u sets: $M^i(1 \leq i \leq u)$. Each of M^i contains v machines. For convenience, we denote by $M^i_j(1 \leq j \leq v)$ the machines in $M^i(i = 1, 2, \ldots, u)$. If all machines in M^i are available, we call M^i available.

Denote by $A(t)$ the set containing all jobs that have arrived at or before time t and that have not been started by time t. We partition $A(t)$ into v sets: $A_i(t) = \{J_j | \frac{i-1}{v} p(A(t)) \leq p_j < \frac{i}{v} p(A(t)), J_j \in A(t)\}, i = 1, 2, \ldots, v-1$ and $A_v(t) = \{J_j | p_j \geq (1 - \frac{1}{v}) p(A(t)), J_j \in A(t)\}$.

Algorithm $H(u, v)$ Whenever there exists an available machine group $M^i(i = 1, 2, \ldots, u)$ and $A(t) \neq \emptyset$, make decision as following. If $t \geq (1 + \frac{1}{u})r(A(t)) + \frac{1}{u}p(A(t))$, start $A_j(t)$ as a batch on M^i_j for each $j(1 \leq j \leq v)$. Otherwise, do nothing but wait.

For convenience, denote by B^x the set of the v batches with the same starting time. Each of the v batches is denoted by B^x_y and B^x_y consists of all jobs in $A_y(t)$. We call B^x a batch group.

Denote by σ the schedule produced by $H(u, v)$. For convenience, denote by $S(B^i), C(B^i)$ the starting time and the maximum completion time of the batches in B^i. If $S(B^i) = (1 + \frac{1}{u})r(B^i) + \frac{1}{u}p(B^i)$, we say that B^i is regular.

Property 2 *For any two batch groups in* σ, *say* B^i *and* B^j, *if* $S(B^i) < S(B^j)$, *then* $r(B^j) > S(B^i)$ *and* $S(B^j) \geq (1 + \frac{1}{u})r(B^j)$.

Lemma 5 *In σ, any batch group B^i is regular, i.e. $S(B^i) = (1+\frac{1}{u})r(B^i) + \frac{1}{u}p(B^i)$.*

Proof Suppose to the contrary that B^i is not regular, i.e. $S(B^i) > (1+\frac{1}{u})r(B^i) + \frac{1}{u}p(B^i)$. Thus between $(1+\frac{1}{u})r(B^i)+\frac{1}{u}p(B^i)$ and $S(B^i)$, there is a batch group which is processed on each machine set $M^j (1 \le j \le u)$. Suppose the u batch groups are $B^{i_j} (1 \le j \le u)$ with $S(B^{i_1}) < S(B^{i_2}) < \cdots < S(B^{i_u})$. Thus, for each $j (1 \le j \le u)$, $S(B^{i_j}) + p(B^{i_j}) \ge S(B^i)$. Hence,

$$S\big(B^i\big) \le \frac{1}{u}\sum_{j=1}^{u}\big(S\big(B^{i_j}\big) + p\big(B^{i_j}\big)\big) = \frac{1}{u}\sum_{j=1}^{u} C\big(B^{i_j}\big) \tag{12}$$

By the algorithm and Property 2, we have

$$S\big(B^{i_1}\big) \ge \frac{1}{u}p\big(B^{i_1}\big) \tag{13}$$

$$\begin{aligned} S\big(B^{i_j}\big) &\ge \left(1+\frac{1}{u}\right) r\big(B^{i_j}\big) + \frac{1}{u}p\big(B^{i_j}\big) \\ &> \left(1+\frac{1}{u}\right) S\big(B^{i_{j-1}}\big) + \frac{1}{u}p\big(B^{i_j}\big) \end{aligned} \tag{14}$$

Thus

$$\sum_{j=1}^{u} S\big(B^{i_j}\big) \ge \left(1+\frac{1}{u}\right)\sum_{j=1}^{u-1} S\big(B^{i_j}\big) + \frac{1}{u}\sum_{j=1}^{u} p\big(B^{i_j}\big)$$

i.e.

$$S\big(B^{i_u}\big) \ge \frac{1}{u}\sum_{j=1}^{u-1} S\big(B^{i_j}\big) + \frac{1}{u}\sum_{j=1}^{u} p\big(B^{i_j}\big) \tag{15}$$

Recall that $S(B^i) > \big(1+\frac{1}{u}\big)S(B^{i_u})$. We have that

$$\begin{aligned} S\big(B^i\big) &> S\big(B^{i_u}\big) + \frac{1}{u}S\big(B^{i_u}\big) \\ &\ge \frac{1}{u}\sum_{j=1}^{u-1} S\big(B^{i_j}\big) + \frac{1}{u}\sum_{j=1}^{u} p\big(B^{i_j}\big) + \frac{1}{u}S\big(B^{i_u}\big) \\ &= \frac{1}{u}\sum_{j=1}^{u}\big(S\big(B^{i_j}\big) + p\big(B^{i_j}\big)\big) \end{aligned} \tag{16}$$

which contradicts the inequality (12).

Therefore, in σ, each batch group B^i is regular, i.e. $S(B^i) = (1 + \frac{1}{u})r(B^i) + \frac{1}{u}p(B^i)$. □

Theorem 2 *The competitive ratio of Algorithm $H(u, v)$ is $1 + \frac{1}{u} + \frac{1}{v}$.*

Proof Let π be an optimal schedule.

In σ, let J_k be the first job that assumes the objective value $L_{\max}(\sigma) = L_k(\sigma)$. Suppose that J_k belongs to batch B_i^l. Since B^l is regular and $p(B_i^l) \leq \frac{i}{v}p(B^l)$,

$$\begin{aligned} L_{max}(\sigma) &= S\left(B^l\right) + p\left(B_i^l\right) + q_k \\ &\leq \left(1 + \frac{1}{u}\right) r\left(B^l\right) + \frac{1}{u} p\left(B^l\right) + \frac{i}{v} p\left(B^l\right) + q_k \end{aligned} \tag{17}$$

While in the optimal schedule π,

$$L_{max}(\pi) \geq r\left(B^l\right) + p_k + q_k \geq r\left(B^l\right) + \frac{i-1}{v} p\left(B^l\right) + q_k \tag{18}$$

$$L_{max}(\pi) \geq r\left(B^l\right) + p\left(B^l\right) \tag{19}$$

Thus it is from the inequality (18)+($\frac{1}{u} + \frac{1}{v}$)·(19) and (17) that $L_{max}(\sigma) \leq (1 + \frac{1}{u} + \frac{1}{v})L_{max}(\pi)$.

Hence, Algorithm $H(u, v)$ is $(1 + \frac{1}{u} + \frac{1}{v})$-competitive.

Now, we can present an instance to show that the bound is tight. Let $m = uv \geq 2$, where u, v are integers. Suppose there are two jobs: $J_1(r_1 = 0, p_1 = 1, q_1 = 0)$, $J_2(r_2 = 0, p_2 = 1 - 1/v, q_2 = 1/v)$. The algorithm will schedule J_1 and J_2 as a single batch which starts at $1/u$. Thus the object value will be $1 + 1/u + 1/v$. While the optimal schedule will assign J_1 and J_2 in different batches with starting time 0. Then the optimal value is 1. Hence the bound is tight.

This completes the proof. □

Acknowledgments This work was supported by the National Nature Science Foundation of China (11101147, 11371137) and the Fundamental Research Funds for the Central Universities.

References

Deng XT, Poon CK, Zhang YZ (2003) Approximation algorithms in batch processing. J Comb Optim 7:247–257

Fang Y, Lu X, Liu P (2011) Online batch scheduling on parallel machines with delivery times. Theor Comput Sci 412:5333–5339

Graham RL, Lawer EL, Lenstra JK (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann Discr Math 5:287–326

Hall LA, Shmoys DB (1989) Approximation schemes for constrained scheduling problems. In: Proceedings of the 30th Annual Symposium on Foundations of Computer Science 134–139

Hoogeveen JA, Vestjean APA (2000) A best possible deterministic online algorithm for minimizing maximum delivery times on a single machine. SIAM J Discr Math 13:56–63

Lee CY, Uzsoy R, Martin-Vega LA (1992) Efficient algorithms for scheduling semi-conductor burn-in operations. Oper Res 40:764–775

Lee CY, Uzsoy R (1999) Minimizing makespan on a single batch processing machine with dynamic job arrivals. Int J Prod Res 37:219–236
Liu P, Lu X, Fang Y (2012) A best possible deterministic online algorithm for minimizing makespan on parallel batch machines. J Sched 15:77–81
Liu ZH, Yu WC (2000) Scheduling one batch processor subject to job release dates. Discr Appl Math 105:129–136
Nong QQ, Cheng TCE, Ng CT (2008) An improved on-line algorithm for scheduling on two unresrtictive paralle batch processing machines. Oper Res Lett 36:584–588
Poon CK, Yu WC (2005) On-line scheduling algorithms for a batch machine with finite capacity. J Comb Optim 9:167–186
Tian J, Fu R, Yuan J (2007) Online scheduling with delivery time on a single batch machine. Theor Comput Sci 374:49–57
Tian J, Cheng TCE, Ng CT, Yuan J (2009) Online scheduling on unbounded parallel-batch machines to minimize the makespan. Inf Process Lett 109:1211–1215
Tian J, Cheng TCE, Ng CT, Yuan J (2012) An improved on-line algorithm for single parallel-batch machine scheduling with delivery times. Discr Appl Math 160:1191–1210
Vestjens APA (1997) Online machine scheduling. Ph.D. Dissertation, Department of mathematics and Computing Science, Eindhoven University of Techology, Eindhoven, The Netherlands
Yuan J, Li S, Tian J, Fu R (2009) A best on-line algorithm for the single machine parallel-batch scheduling with restricted delivery times. J Comb Optim 17:206–213
Zhang G, Cai X, Wong CK (2001) On-line algorithms for minimizing makespan on batch processing machines. Nav Res Logist 48:241–258