Theoretical Computer Science 410 (2009) 2291–2294

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Note

A best online algorithm for scheduling on two parallel batch machines ☆

Ji Tian, Ruyan Fu, Jinjiang Yuan *

Department of Mathematics, Zhengzhou University, Zhengzhou, Henan 450052, People's Republic of China

ARTICLE INFO

Article history:
Received 29 April 2008
Received in revised form 4 February 2009
Accepted 14 February 2009
Communicated by D.-Z. Du

Keywords:
Online scheduling
Parallel batch machines
Competitive ratio

ABSTRACT

We consider the online scheduling on two parallel batch machines with infinite batch size to minimize makespan, where jobs arrive over time. That is, all information of a job is not available until it is released. For this online scheduling problem, Nong et al. [Q.Q. Nong, T.C.E. Cheng, C.T. Ng, An improved online algorithm for scheduling on two unrestrictive parallel batch processing machines, Operations Research Letters, 36 (2008) 584–588] have provided an online algorithm with competitive ratio no greater than $\sqrt{2}$. We show that this bound is tight for the problem. Furthermore we give a new best possible online algorithm with a tighter structure.

1. Introduction

Online scheduling, including online over list and online over time, has been extensively studied for a long time. The model studied in this paper is an online over time system. That is, jobs arrive over time, and the characteristics of each job, including arrival time, processing time and so on, are unknown until its arrival time. For a job J_i, its arrival time and processing time are denoted by r_i and p_i, respectively. In parallel batch scheduling, a machine can process b jobs simultaneously as a batch with capacity b. Jobs in a batch have a common processing time and a common completion time. The processing time of a batch is defined to be the maximum processing time of the jobs in the batch. We say that the batch size is infinite if $b = \infty$.

The qualities of online algorithms are assessed by their competitive ratio. An algorithm is called ρ-competitive if, for any instance, a solution is achieved with value not worse than ρ times the value of an optimal off-line solution.

For online scheduling on m identical machines to minimize makespan, Chen and Vestjens [1] proposed an online *LPT* (largest processing time) algorithm with competitive ratio 3/2, and proved that any online algorithm has a competitive ratio of at least 1.3473, while for $m = 2$, the lower bound is $(5 - \sqrt{5})/2$. When $m = 2$, Noga and Seiden [3] provided a best possible online algorithm with competitive ratio $(5 - \sqrt{5})/2$. In the off-line version, for scheduling n jobs with release dates on m identical parallel batch machines to minimize the total weighted completion times of the jobs, Li et al. [2] presented a polynomial-time approximation scheme (*PTAS*). In online scheduling, Zhang et al. [6] considered the problem of the equal size jobs on m identical parallel batch machines to minimize makespan. When the batch size is infinite, they proposed a best possible online algorithm with competitive ratio $1+\beta_m$, where β_m is the positive solution of equation $(1+\beta_m)^{m+1} = \beta_m+2$. When the batch size is finite, they provided a best possible online algorithm with competitive ratio $(\sqrt{5} + 1)/2$.

In this paper, we consider the online scheduling on two parallel batch machines with infinite capacity to minimize makespan. For this problem, Nong et al. [4] have provided an online algorithm with competitive ratio no greater than $\sqrt{2}$. We show in this paper that the lower bound of competitive ratio of the problem is $\sqrt{2}$. This means that the online algorithm provided by Nong et al. is the best possible. Then we propose another best possible online algorithm with competitive ratio

☆ Research supported by NSFC (10671183), NFSC-RGC (70731160633) and SRFDP (20070459002).

* Corresponding author. Tel.: +86 371 67767835.
E-mail address: yuanjj@zzu.edu.cn (J. Yuan).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2009.02.011

$\sqrt{2}$. The new algorithm, called Modified-Sleepy, has a tighter structure, which enables us to give a simple proof for the competitive ratio. As for the idea to tighten the structure of the previous algorithm, it is similar to that of Poon and Yu [5].

The paper is organized as follows. In Section 2, we prove that, for any online algorithm, the lower bound of competitive ratio is no less than $\sqrt{2}$. In Section 3, we propose a best possible online algorithm called Modified-Sleepy, since we adopt the advantages of the Sleepy algorithm provided by Noga and Seiden [3].

2. Lower bound

In this section we will prove that no online algorithm has a competitive ratio less than $\sqrt{2}$. To find the lower bound of competitive ratio, we use adversary strategy to construct a special instance. Let $\alpha = \sqrt{2} - 1$. Then we have $\alpha^2 + 2\alpha - 1 = 0$. We use $C_{\max}(\sigma)$ and $C_{\max}(\pi)$ to denote the objective value generated by the online algorithm and by an optimal off-line algorithm, respectively.

Theorem 2.1. *For the online scheduling problem on two parallel batch machines with infinite batch size to minimize makespan, there exists no online algorithm with competitive ratio less than* $\sqrt{2}$.

Proof. Consider the following instance provided by the adversary. The first job J_1, with processing time $p_1 = 1$, arrives at time 0. Suppose that the online algorithm starts job J_1 at time S_1 on machine 1.

If $S_1 \geq \alpha$, then $C_{\max}(\sigma) \geq S_1 + 1 \geq 1 + \alpha$ and $C_{\max}(\pi) = 1$. Thus $C_{\max}(\sigma)/C_{\max}(\pi) \geq 1 + \alpha$.

If $S_1 < \alpha$, the second job J_2 with processing time $p_2 = 1 - S_1$ arrives at time $S_1 + \epsilon$. If J_2 is processed on machine 1, then $C_{\max}(\sigma) \geq S_1 + 1 + 1 - S_1 = 2$ and $C_{\max}(\pi) = 1$. Thus $C_{\max}(\sigma)/C_{\max}(\pi) > 1 + \alpha$. So we assume that the online algorithm starts to process J_2 at time S_2 on machine 2. We consider the following three cases.

Case 1. $S_2 \geq S_1 + \alpha$. Then $C_{\max}(\sigma) \geq S_2 + p_2 \geq S_1 + \alpha + 1 - S_1 = 1 + \alpha$ and $C_{\max}(\pi) = 1 + \epsilon$. Thus $C_{\max}(\sigma)/C_{\max}(\pi) \geq 1 + \alpha$ as $\epsilon \longrightarrow 0$.

Case 2. $2S_1 \leq S_2 < S_1 + \alpha$. The third job J_3 with processing time $p_3 = \alpha(S_1 + 1)$ arrives at time $S_2 + \epsilon$. Since the completion time of J_2 is $S_2 + 1 - S_1 \geq 2S_1 + 1 - S_1 = S_1 + 1$ and the completion time of J_1 is equal to $S_1 + 1$, we know that the starting time of the third job cannot be earlier than time moment $S_1 + 1$. Then $C_{\max}(\sigma) \geq S_1 + 1 + \alpha(S_1 + 1) = (1 + \alpha)(S_1 + 1)$. Since there exists a feasible schedule, say π', in which J_1 and J_2 are processed in a common batch on a machine and J_3 is processed on the other machine, we have $C_{\max}(\pi) \leq C_{\max}(\pi') = \max\{S_1 + \epsilon + 1, S_2 + \epsilon + \alpha(S_1 + 1)\}$. By the fact that $S_2 + \epsilon + \alpha(S_1 + 1) \leq S_1 + \alpha + \epsilon + \alpha(S_1 + 1) = S_1 + \epsilon + \alpha(S_1 + 2) \leq S_1 + \epsilon + \alpha(\alpha + 2) = S_1 + 1 + \epsilon$ (recall that $S_1 < \alpha$), we conclude $C_{\max}(\sigma)/C_{\max}(\pi) \geq 1 + \alpha$ as $\epsilon \longrightarrow 0$.

Case 3. $S_1 + \epsilon = r_2 \leq S_2 < 2S_1$. The third job J_3 with processing time $p_3 = 1 + S_1 - S_2$ arrives at time $S_2 + \epsilon$. In this case the completion time of the second job is $S_2 + 1 - S_1 < 2S_1 + 1 - S_1 = S_1 + 1$, which is earlier than the completion time of the first job and it is also the earliest possible time moment for starting job J_3. Then we have $C_{\max}(\sigma) \geq S_2 + 1 - S_1 + 1 + S_1 - S_2 = 2$ and $C_{\max}(\pi) \leq \max\{S_1 + \epsilon + 1, S_2 + \epsilon + 1 + S_1 - S_2\} = S_1 + \epsilon + 1$ (since there exists a feasible schedule in which J_1 and J_2 are processed in a common batch on a machine and J_3 is processed on the other machine). Thus $C_{\max}(\sigma)/C_{\max}(\pi) \geq 2/(S_1 + \epsilon + 1) \geq 2/(\alpha + \epsilon + 1) \longrightarrow 1 + \alpha$ as $\epsilon \longrightarrow 0$, since $S_1 < \alpha$. This completes the proof of Theorem 2.1. □

3. A new online algorithm

Since the algorithm is non-preemptive, when two machines are busy, it should wait until at least one machine is idle. So it needs to properly handle the case when two machines are idle and the case when one machine is running (busy) and the other one is idle. Note that the capacity of batch is unbounded. At each starting time of batches, the algorithm will process all unscheduled available jobs as a single batch. That is, at a time moment, there is at most one batch starting to process.

Some notations will be used in the algorithm. Let $U(t)$ denote the set of all unscheduled available jobs at time t. We say $J(t)$ is the last longest job in $U(t)$ means that it has the largest arrival time among all longest jobs in $U(t)$. Let $p(t)$ and $r(t)$ denote the processing time and arrival time of job $J(t)$, respectively. If at time t, only one machine is running a batch, we use $B^*(t)$ to denote this batch, and suppose that it has starting time $S^*(t)$ and processing time $p^*(t)$. If at time t, both machines are idle, we define $S^*(t) = p^*(t) = 0$. Let $\alpha = \sqrt{2} - 1$, which is the positive solution of equation $\alpha^2 + 2\alpha - 1 = 0$.

First of all, we recall the online algorithm $A_2(\alpha)$ provided by Nong et al. [4] in order to compare to the Modified-Sleepy algorithm in this paper.

Algorithm $A_2(\alpha)$**.** At time t, if a machine is idle, $U(t) \neq \emptyset$, and $t \geq (1 + \alpha)r(t) + \alpha p(t)$, then start $U(t)$ as a single batch on the machine at time t; otherwise, do nothing but wait.

By deliberating the structure of the instance in the proof of Theorem 2.1 and taking advantage of the Sleepy algorithm [3], the Modified-Sleepy algorithm is designed as follows.

Modified-Sleepy **Algorithm:**

At time t, if a machine is idle, $U(t) \neq \emptyset$, and $t \geq \max\{\alpha p(t), S^*(t) + \alpha p^*(t)\}$, then start $U(t)$ as a single batch on the machine; otherwise, do nothing but wait.

It can be observed that, at each decision time t, the action of algorithm $A_2(\alpha)$ is uniquely determined by the information of the last longest job in $U(t)$ without considering the running batch $B^*(t)$ at time t. This is the main reason for the complicated proof in [4]. In the Modified-Sleepy algorithm, the action at each decision time t is additionally affected by the running batch $B^*(t)$. Hence, the strategy in the Modified-Sleepy algorithm seems more reasonable. Although both algorithms apply delay strategy, the delay caused in the Modified-Sleepy algorithm is less than that in algorithm $A_2(\alpha)$ in most cases. Hence, the Modified-Sleepy algorithm has a tighter structure than algorithm $A_2(\alpha)$. This enables us to give a simple proof.

We use M_1 and M_2 to denote the two parallel batch machines. Let σ and π denote the schedule generated by the Modified-Sleepy algorithm and an optimal off-line schedule, respectively. Their objective values are denoted by $C_{\max}(\sigma)$ and $C_{\max}(\pi)$, respectively. We assume that there are n batches totally in σ, which are written as $B_1, B_2, \ldots, B_n$. For each i, the last longest job in batch B_i is denoted by J_i with processing time p_i and arrive time r_i. Let S_i and C_i be the starting time and the completion time of batch B_i, respectively. Suppose that $C_1 \le C_2 \le \cdots \le C_n$. It can be observed that $S_i < S_j$ implies $r_j > S_i$. The objective value $C_{\max}(\sigma)$ is assumed by batch B_n, i.e., $C_{\max}(\sigma) = C_n$. Without loss of generality, we assume that batch B_n is scheduled on machine M_1.

In schedule σ, if there exists some batch with the starting time greater than S_n, we will cancel this batch, since the objective value is reached by batch B_n. This would not impact the value of $C_{\max}(\sigma)$, but may possibly decrease that of $C_{\max}(\pi)$. The competitive ratio would not decrease. So we suppose in the sequel that no jobs arrive after time moment S_n. Let us start by giving a claim that was first proposed in Nong et al. [4].

Claim 3.1 ([4]). *Without decreasing the ratio of $C_{\max}(\sigma)/C_{\max}(\pi)$, we can assume that there is only one job in each batch of σ.* □

By Claim 3.1, we use the set of n jobs $\{J_1, \ldots, J_n\}$ to replace the set of n batches $\{B_1, \ldots, B_n\}$ generated by the Modified-Sleepy algorithm. We say job J_i is running at time t if batch B_i is running at that time moment in σ. We consider the last three jobs J_{n-2}, J_{n-1} and J_n. Note that $C_{n-2} \le C_{n-1} \le C_n$ and J_n is scheduled on machine M_1 in σ. Without loss of generality, we suppose that J_{n-2} is also scheduled on M_1 and J_{n-1} is scheduled on machine M_2. In the following, we will offer some claims that describe the structure of σ.

Claim 3.2. *For each job J_i, with $1 \le i \le n$, the starting time S_i satisfies $S_i \ge \alpha p_i$.* □

Claim 3.3. *If there exists a job J_i in σ such that $C_{\max}(\sigma) - C_{\max}(\pi) \le (1-\alpha)p_i$ and $C_{\max}(\pi) \ge S_i + p_i$ or $C_{\max}(\pi) \ge (1+\alpha)p_i$, then we have $C_{\max}(\sigma)/C_{\max}(\pi) \le 1 + \alpha$.*

Proof. From Claim 3.2, we have that $S_i \ge \alpha p_i$. Then the inequality $C_{\max}(\pi) \ge S_i + p_i$ implies that $C_{\max}(\pi) \ge (1+\alpha)p_i$. By the fact that $C_{\max}(\sigma) - C_{\max}(\pi) \le (1-\alpha)p_i$, we conclude that $(C_{\max}(\sigma) - C_{\max}(\pi))/C_{\max}(\pi) \le (1-\alpha)/(1+\alpha) = \alpha$. The result follows. □

A job J_i is called *normal* in σ if the starting time $S_i = \max\{S^*(S_i) + \alpha p^*(S_i), \alpha p_i, r_i\}$. If job J_i is not normal in σ, then, for every job J_j with $S_j < S_i$, we have $S_i > S_j + \alpha p_j$.

Claim 3.4. *If J_n is a normal job in σ, then we have $C_{\max}(\sigma)/C_{\max}(\pi) \le 1 + \alpha$.*

Proof. If there exists a running job at time moment S_n in σ, it must be job J_{n-1}. Since J_n is a normal job in σ, we have $S_n = \max\{S_{n-1} + \alpha p_{n-1}, \alpha p_n, r_n\}$. Note that $r_n > S_{n-1}$. Then we have $C_{\max}(\pi) > S_{n-1} + p_n$. If $S_n = S_{n-1} + \alpha p_{n-1}$, then $C_{\max}(\sigma) = S_{n-1} + \alpha p_{n-1} + p_n$ and so $C_{\max}(\sigma) - C_{\max}(\pi) < \alpha p_{n-1} < \alpha C_{\max}(\pi)$. If $S_n = \max\{\alpha p_n, r_n\}$, then $C_{\max}(\sigma) = S_n + p_n = \max\{(1+\alpha)p_n, r_n + p_n\} \le (1+\alpha)C_{\max}(\pi)$. The result follows. □

In the following, we suppose that J_n is not a normal job in σ. Then we have $S_n = C_{n-2} > \max\{S_{n-1} + \alpha p_{n-1}, \alpha p_n, r_n\}$, and $C_{\max}(\sigma) = S_{n-2} + p_{n-2} + p_n$.

Claim 3.5. *If, for a certain $i \in \{n-1, n-2\}$, J_i is a normal job and $C_{\max}(\pi) \ge r_i + p_i + p_n$, then $C_{\max}(\sigma)/C_{\max}(\pi) \le 1 + \alpha$.*

Proof. Suppose that the running job (if exists) is J^* at time moment S_i. Then $r_i > S^*$ and $S_i = \max\{S^* + \alpha p^*, \alpha p_i, r_i\}$, since job J_i is normal in σ. When $S_i = S^* + \alpha p^*$, we have $C_{\max}(\sigma) \le S^* + \alpha p^* + p_i + p_n$. Recall that $C_{\max}(\pi) \ge r_i + p_i + p_n > S^* + p_i + p_n$. Then $C_{\max}(\sigma) - C_{\max}(\pi) < \alpha p^* < \alpha C_{\max}(\pi)$. When $S_i = \max\{\alpha p_i, r_i\}$, since $C_{\max}(\pi) \ge r_i + p_i + p_n$, we have $C_{\max}(\sigma) - C_{\max}(\pi) < \alpha p_i < \alpha C_{\max}(\pi)$. The result follows. □

In the following discussion, we suppose that $\{S_{n-1}, S_{n-2}\} = \{S_{i_1}, S_{i_2}\}$ such that $S_{i_1} > S_{i_2}$. Then $C_{\max}(\sigma) = S_{n-2} + p_{n-2} + p_n \le S_{n-1} + p_{n-1} + p_n$, and therefore, $C_{\max}(\sigma) \le \min\{S_{i_1} + p_{i_1} + p_n, S_{i_2} + p_{i_2} + p_n\}$.

Theorem 3.6. $C_{\max}(\sigma)/C_{\max}(\pi) \le 1 + \alpha$.

Proof. By the implementation of the Modified-Sleepy algorithm and the fact that $S_{i_1} > S_{i_2}$, we have $S_{i_1} \ge S_{i_2} + \alpha p_{i_2}$ and $r_n > S_{i_1} \ge r_{i_1} > S_{i_2}$. Then $C_{\max}(\pi) \ge r_n + p_n > S_{i_2} + \alpha p_{i_2} + p_n$. Using the fact that $C_{\max}(\sigma) \le S_{i_2} + p_{i_2} + p_n$, we have $C_{\max}(\sigma) - C_{\max}(\pi) \le (1-\alpha)p_{i_2}$.

If job J_{i_2} starts at or after time moment αp_{i_2} in schedule π, then we have $C_{\max}(\pi) \ge (1+\alpha)p_{i_2}$. From Claim 3.3, we conclude that $C_{\max}(\sigma)/C_{\max}(\pi) \le 1 + \alpha$.

We suppose below that J_{i_2} starts before time moment αp_{i_2} in schedule π. By Claim 3.2, we know that $S_{i_2} \geq \alpha p_{i_2}$. With the fact that $r_{i_1} > S_{i_2}$ and $r_n > S_{i_2}$, we conclude that neither of the jobs J_{i_1} and J_n belong to a common batch with job J_{i_2} in schedule π. We analyze this situation by the following two cases, keeping in mind that $C_{\max}(\sigma) - C_{\max}(\pi) \leq (1-\alpha)p_{i_2}$.

Case 1. Jobs J_{i_1} and J_n are not scheduled on the same machine in schedule π. Then we have $C_{\max}(\pi) \geq \min\{r_{i_2} + p_{i_2} + p_{i_1}, r_{i_2} + p_{i_2} + p_n\}$. If $C_{\max}(\pi) \geq r_{i_2} + p_{i_2} + p_{i_1}$, we distinguish two subcases. When $p_{i_1} \geq \alpha p_{i_2}$, we have $C_{\max}(\pi) \geq (1+\alpha)p_{i_2}$. Note that $C_{\max}(\sigma) - C_{\max}(\pi) \leq (1-\alpha)p_{i_2}$. By Claim 3.3, the result $C_{\max}(\sigma)/C_{\max}(\pi) \leq 1+\alpha$ holds. When $p_{i_1} < \alpha p_{i_2}$, since $C_{\max}(\sigma) \leq S_{i_1} + p_{i_1} + p_n$ and $C_{\max}(\pi) \geq r_n + p_n > S_{i_1} + p_n$, we have $C_{\max}(\sigma) - C_{\max}(\pi) \leq p_{i_1} < \alpha p_{i_2} < \alpha C_{\max}(\pi)$.

If $C_{\max}(\pi) \geq r_{i_2} + p_{i_2} + p_n$, from Claim 3.5, we need to properly handle the case that job J_{i_2} is not normal. Then S_{i_2} is the completion time of some batch in schedule σ. Assume that $S_{i_2} = C_{i_3}$ and job J_{i_4} is running or completes at time moment S_{i_2}. Then $C_{i_3} \leq C_{i_4}$. Further, in σ job J_{i_3} is scheduled on the same machine as J_{i_2} and job J_{i_4} is on the same machine as J_{i_1}. Thus $S_{i_1} \geq C_{i_4}$. We need to consider two possibilities depending on the relationship of S_{i_3} and S_{i_4}.

First, $S_{i_4} > S_{i_3}$. Then we have $r_{i_2} > S_{i_4}$ and $S_{i_4} \geq S_{i_3} + \alpha p_{i_3}$. Since $C_{\max}(\pi) \geq r_{i_2} + p_{i_2} + p_n \geq S_{i_3} + \alpha p_{i_3} + p_{i_2} + p_n$, by the fact that $C_{\max}(\sigma) \leq S_{i_3} + p_{i_3} + p_{i_2} + p_n$, we have $C_{\max}(\sigma) - C_{\max}(\pi) \leq (1-\alpha)p_{i_3}$. Note that $C_{\max}(\pi) > r_n > S_{i_2} = S_{i_3} + p_{i_3}$. Using Claim 3.3, we conclude that $C_{\max}(\sigma)/C_{\max}(\pi) \leq 1+\alpha$.

Second, $S_{i_4} < S_{i_3}$. Then we have $r_{i_2} > S_{i_3}$ and $S_{i_3} \geq S_{i_4} + \alpha p_{i_4}$. Recall that $S_{i_3} + p_{i_3} \leq S_{i_4} + p_{i_4}$. Then we have $p_{i_4} - p_{i_3} \geq S_{i_3} - S_{i_4} \geq \alpha p_{i_4}$, and so, $p_{i_3} \leq (1-\alpha)p_{i_4}$. This implies that $C_{\max}(\sigma) - C_{\max}(\pi) \leq (S_{i_3} + p_{i_3} + p_{i_2} + p_n) - (S_{i_3} + p_{i_2} + p_n) = p_{i_3} \leq (1-\alpha)p_{i_4}$. Since $C_{\max}(\pi) > r_n > S_{i_1} \geq S_{i_4} + p_{i_4}$, by Claim 3.3, we conclude that $C_{\max}(\sigma)/C_{\max}(\pi) \leq 1+\alpha$.

Case 2. Jobs J_{i_1} and J_n are scheduled on a common machine in schedule π. Note that $r_n > S_{i_1} \geq r_{i_1} > S_{i_2}$. Depending on whether J_{i_1} and J_n belong to a common batch or not in π, we consider two possibilities.

First, jobs J_{i_1} and J_n belong to a common batch in π. Since $r_n > S_{i_1} \geq S_{i_2} + \alpha p_{i_2}$, we have $C_{\max}(\pi) \geq S_{i_2} + \alpha p_{i_2} + \max\{p_{i_1}, p_n\}$. If $\max\{p_{i_1}, p_n\} \geq (1-\alpha)p_{i_2}$, by using Claim 3.2, we have $C_{\max}(\pi) \geq S_{i_2} + \alpha p_{i_2} + \max\{p_{i_1}, p_n\} \geq (1+\alpha)p_{i_2}$. From the fact that $C_{\max}(\sigma) - C_{\max}(\pi) \leq (1-\alpha)p_{i_2}$ and Claim 3.3, we conclude that $C_{\max}(\sigma)/C_{\max}(\pi) \leq 1+\alpha$. Then we assume that $\max\{p_{i_1}, p_n\} < (1-\alpha)p_{i_2}$. Note that $C_{\max}(\sigma) - C_{\max}(\pi) \leq (S_{i_1} + p_{i_1} + p_n) - (S_{i_1} + \max\{p_{i_1}, p_n\}) < \min\{p_{i_1}, p_n\} \leq \max\{p_{i_1}, p_n\}$. Thus

$$\frac{C_{\max}(\sigma) - C_{\max}(\pi)}{C_{\max}(\pi)} \leq \frac{\max\{p_{i_1}, p_n\}}{S_{i_2} + \alpha p_{i_2} + \max\{p_{i_1}, p_n\}} \leq \frac{(1-\alpha)p_{i_2}}{\alpha p_{i_2} + \alpha p_{i_2} + (1-\alpha)p_{i_2}} \leq \frac{1-\alpha}{1+\alpha} = \alpha.$$

Second, J_{i_1} and J_n belong to distinct batches in π. Then $C_{\max}(\pi) \geq r_{i_1} + p_{i_1} + p_n$. If J_{i_1} is a normal job in σ, from Claim 3.5, the result $C_{\max}(\sigma)/C_{\max}(\pi) \leq 1+\alpha$ holds. Suppose that J_{i_1} is not a normal job in σ. Then S_{i_1} is the completion time of some batch in schedule σ. Suppose $S_{i_1} = S_{i_3} + p_{i_3}$. We consider the relationship of S_{i_2} and S_{i_3} by the following two subcases.

If $S_{i_2} > S_{i_3}$, then $S_{i_2} \geq S_{i_3} + \alpha p_{i_3}$. Further, with the fact that $r_{i_1} > S_{i_2}$, we have $C_{\max}(\pi) \geq S_{i_2} + p_{i_1} + p_n \geq S_{i_3} + \alpha p_{i_3} + p_{i_1} + p_n$. Thus $C_{\max}(\sigma) \leq S_{i_1} + p_{i_1} + p_n = S_{i_3} + p_{i_3} + p_{i_1} + p_n$, and so, $C_{\max}(\sigma) - C_{\max}(\pi) \leq (1-\alpha)p_{i_3}$. From the fact that $C_{\max}(\pi) > r_n > S_{i_1} = S_{i_3} + p_{i_3}$ and Claim 3.3, we have $C_{\max}(\sigma)/C_{\max}(\pi) \leq 1+\alpha$.

If $S_{i_2} < S_{i_3}$, then $r_{i_1} > S_{i_3} \geq S_{i_2} + \alpha p_{i_2}$ and $C_{\max}(\sigma) - C_{\max}(\pi) \leq (S_{i_3} + p_{i_3} + p_{i_1} + p_n) - (S_{i_3} + p_{i_1} + p_n) = p_{i_3}$. From the fact that $S_{i_3} + p_{i_3} \leq S_{i_2} + p_{i_2}$, we have $p_{i_3} < (1-\alpha)p_{i_2}$. Note that $C_{\max}(\pi) > r_n > S_{i_3} + p_{i_3} \geq S_{i_2} + \alpha p_{i_2} + p_{i_3}$. We conclude that

$$\frac{C_{\max}(\sigma) - C_{\max}(\pi)}{C_{\max}(\pi)} \leq \frac{p_{i_3}}{S_{i_2} + \alpha p_{i_2} + p_{i_3}} \leq \frac{(1-\alpha)p_{i_2}}{\alpha p_{i_2} + \alpha p_{i_2} + (1-\alpha)p_{i_2}} = \frac{1-\alpha}{1+\alpha} = \alpha.$$

This completes the proof of Theorem 3.6. $\square$

References

[1] B. Chen, A.P.A. Vestjens, Scheduling on identical machines: How good is LPT in an on-line setting? Operations Research Letters 21 (1997) 165–169.

[2] S.G. Li, G.J. Li, X.Q. Qi, Minimizing total weighted completion time on identical parallel batch machines, International Journal of Foundations of Computer Science 176 (2006) 1441–1454.

[3] J. Noga, S.S. Seiden, An optimal online algorithm for scheduling two machines with release times, Theoretical Computer Science 268 (2001) 133–143.

[4] Q.Q. Nong, T.C.E. Cheng, C.T. Ng, An improved on-line algorithm for scheduling on two unrestrictive parallel batch processing machines, Operations Research Letters 36 (2008) 584–588.

[5] C.K. Poon, W.C. Yu, A flexible on-line scheduling algorithm for batch machine with infinite capacity, Annals of Operations Research 133 (2005) 175–181.

[6] G.C. Zhang, X.Q. Cai, C.K. Wong, Optimal on-line algorithms for scheduling on parallel batch processing machines, IIE Transactions 35 (2003) 175–181.