

Better Algorithm of Ordinal Online Schedule for Jobs with Similar Sizes on Two Machines

Limin Wang¹, Rongheng Li^{1*}, Yunxia Zhou²

¹Key Laboratory of Computing and Stochastic Mathematics (Ministry of Education), Department of Mathematics, School of Mathematics and Statistics, Hunan Normal University, Changsha, China

²College of Information Science and Engineering, Hunan Normal University, Changsha, China

Email: *lirongheng@hunnu.edu.cn

How to cite this paper: Wang, L.M., Li, R.H. and Zhou, Y.X. (2019) Better Algorithm of Ordinal Online Schedule for Jobs with Similar Sizes on Two Machines. *American Journal of Operations Research*, 9, 235-243.

<https://doi.org/10.4236/ajor.2019.95015>

Received: June 5, 2019

Accepted: September 17, 2019

Published: September 20, 2019

Copyright © 2019 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Ordinal online schedule for jobs with similar sizes in $[1, r]$ ($1 \leq r \leq 2$) on two parallel machines system is considered. Firstly it is proved that the worst case performance ratio of the existing algorithm P_2 cannot be improved even if the job processing times are known in $[1, r]$ for any $r \geq 1$. Then a better algorithm named S is developed and its worst case performance ratio is given for $1 \leq r \leq 2$.

Keywords

Semi-Online Scheduling, P_m Algorithm, S Algorithm, Worst Performance Ratio

1. Introduction

The scheduling problem on m parallel identical machines is defined as follows: Given a job set $L = \{J_1, J_2, \dots, J_n\}$ of n jobs where job J_j has non-negative processing time p_j , assign the jobs on m machines $\{M_1, M_2, \dots, M_m\}$ so as to minimize the maximum completion times of the jobs on each machine. The earliest algorithm for on-line scheduling jobs on parallel machines is the List Scheduling (LS) algorithm, which was introduced by Graham [1]. Many models and algorithms for online scheduling are proposed later on. In classic scheduling problem, there is no constraints on the size of job. However, in practice, the size of job can neither be too large nor too small. This motivates researchers to study scheduling problems when the sizes of all jobs are known in $[1, r]$ with $r \geq 1$ [2]-[7].

In this paper, we will consider ordinal online scheduling jobs with sizes in

$[1, r](1 \leq r \leq 2)$ on two parallel machines. The model of ordinal online scheduling was proposed by Liu *et al.* [8]. It is assumed that the values of the processing times are unknown, but that the order of the jobs by non-increasing processing time is known, *i.e.*, without loss of generality that $p_1 \geq p_2 \geq \dots \geq p_n$. An algorithm named P_m was developed for the system of m machines and it is proved that the algorithm is the best online algorithm for $m = 2, 3$. In current research, it will be proved that, for $m = 2$, the worst case performance ratio of algorithm P_2 can not be improved even if the sizes of all jobs are known in $[1, r]$ for any $r \geq 1$. Then a better algorithm named S is proposed for $r \in [1, 2]$ and its worst case performance ratio is given.

The rest of the paper is organized as follows. In Section 2, some definitions and the algorithm S and P_2 are given. Section 3 analyzes the competitive ratio of the algorithm S . Finally, some concluding remarks are given in Section 4.

2. Some Definitions and Algorithms

Definition 1. Given m parallel machines, let $L = \{J_1, J_2, \dots, J_n\}$ be any list of jobs. Algorithm A is a heuristic algorithm. Let $C_{\max}^A(L)$ and $C_{\max}^{OPT}(L)$ be the makespan of algorithm A and the makespan of an optimal off-line algorithm respectively. We refer to

$$R(m, A) = \sup_L \frac{C_{\max}^A(L)}{C_{\max}^{OPT}(L)}$$

as the worst case performance ratio of algorithm A .

In the following of this paper, we always assume that the number of machines is two (*i.e.* $m = 2$) and the sizes of job list $L = \{J_1, J_2, \dots, J_n\}$ satisfies $p_1 \geq p_2 \geq \dots \geq p_n$ and $p_j \in [1, r](j = 1, 2, \dots, n, 1 \leq r \leq 2)$ if no specific explanation is given.

Algorithm P_2 [8]. Jobs are assigned to machines as follows:

$$\begin{aligned} M_1 &: \{J_{3k+1} \mid k \geq 0\}, \\ M_2 &: \{J_{3k-1}, J_{3k} \mid k \geq 1\}. \end{aligned}$$

i.e.

$$\begin{aligned} M_1 &: J_1 \quad J_4 \quad J_7 \quad J_{10} \dots \\ M_2 &: J_2 \quad J_3 \quad J_5 \quad J_6 \quad J_8 \quad J_9 \dots \end{aligned}$$

Algorithm S .

Jobs are assigned to machines as follows:

$$\begin{aligned} M_1 &: \{J_1\} \cup \{J_{4k}, J_{4k+1} \mid k \geq 1\}, \\ M_2 &: \{J_{4k-2}, J_{4k-1} \mid k \geq 1\}. \end{aligned}$$

i.e.

$$\begin{aligned} M_1 &: J_1 \quad J_4 \quad J_5 \quad J_8 \quad J_9 \quad J_{12} \dots \\ M_2 &: J_2 \quad J_3 \quad J_6 \quad J_7 \quad J_{10} \quad J_{11} \dots \end{aligned}$$

The two algorithms are the same for assigning the first four jobs. The dif-

ferences are that P_2 assign the first two jobs on M_2 and the third on M_1 for job set $\{J_j \mid j = 3k + 2, 3k + 3, 3k + 4\} (k \geq 1)$. However algorithm S assign the two consecutive job $J_{2k+1}, J_{2k+2} (k \geq 2)$ on two machines alternatively.

In the following, we consider the worst case performance ratio of algorithm P_2 and S . We will show that algorithm S is better than P_2 under the assumption of $p_j \in [1, r]$ for $r \leq 2$.

3. Main Results

Theorem 1. For algorithm P_2 , its worst case performance ratio is $\frac{4}{3}$. Furthermore, its worst case performance ratio can not be improved if $p_j (j = 1, 2, \dots, n)$ satisfy $p_j \in [1, r]$ for any $r \geq 1$.

Proof: The first conclusion is a direct result from Liu et al. [8]. For the second conclusion, consider job list $L = \{J_1, J_2, \dots, J_{6k+4}\}$ satisfying

$$p_1 = p_2 = p_3 = p_4 = r \geq 1, p_5 = p_6 = \dots = p_{6k+4} = 1.$$

By the rules of P_2 , we get

$$\begin{aligned} M_1 : & r \quad r \quad 1 \quad 1 \dots \\ M_2 : & r \quad r \quad 1 \quad 1 \quad 1 \quad 1 \dots \end{aligned}$$

It is obvious that $C_{\max}^{P_2}(L) = 2r + 4k$ and $C_{\max}^{OPT}(L) = 2r + 3k$. Hence

$$\frac{C_{\max}^{P_2}(L)}{C_{\max}^{OPT}(L)} = \frac{2r + 4k}{2r + 3k} \rightarrow \frac{4}{3} (k \rightarrow \infty).$$

In the following of this paper, let $L_i (i = 1, 2)$ to denote the completion time of machine M_i in the schedule assigned by algorithm S .

Lemma 2. Given any job list $L = \{J_1, J_2, \dots, J_n\}$, the following inequality holds

$$\frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} \leq 1 + \frac{r}{2C_{\max}^{OPT}(L)}.$$

Proof: By the rules of S algorithm, we get

$$\begin{aligned} \frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} &= \frac{\max\{L_1, L_2\}}{C_{\max}^{OPT}(L)} \\ &= \frac{L_1 + L_2 + \max\{L_1, L_2\} - \min\{L_1, L_2\}}{2C_{\max}^{OPT}(L)} \\ &\leq 1 + \frac{|L_1 - L_2|}{2C_{\max}^{OPT}(L)}. \end{aligned}$$

That means it is enough to prove $|L_1 - L_2| \leq r$. We consider it according to the four cases of $n = 4k, n = 4k + 1, n = 4k + 2, n = 4k + 3$.

Case 1: $n = 4k$. In this case,

$$L_1 = p_1 + \sum_{j=1}^{k-1} (p_{4j} + p_{4j+1}) + p_{4k}; \quad L_2 = \sum_{j=1}^k (p_{4j-2} + p_{4j-1}).$$

Hence the following inequalities hold:

$$L_1 - L_2 = p_1 + \sum_{j=1}^{k-1} (p_{4j} + p_{4j+1} - p_{4j-2} - p_{4j-1}) + p_{4k} - p_{4k-2} - p_{4k-1} \leq p_1 \leq r,$$

$$L_2 - L_1 = p_2 + p_3 - p_1 + \sum_{j=1}^{k-1} (p_{4j+2} + p_{4j+3} - p_{4j} - p_{4j+1}) - p_{4k} \leq p_2 \leq r.$$

That means $|L_1 - L_2| \leq r$.

Case 2: $n = 4k + 1$. In this case,

$$L_1 = p_1 + \sum_{j=1}^k (p_{4j} + p_{4j+1}), \quad L_2 = \sum_{j=1}^k (p_{4j-2} + p_{4j-3}).$$

Hence the following inequalities hold:

$$L_1 - L_2 = p_1 + \sum_{j=1}^k (p_{4j} + p_{4j+1} - p_{4j-2} - p_{4j-1}) \leq p_1 \leq r,$$

$$L_2 - L_1 = p_2 + p_3 - p_1 + \sum_{j=1}^{k-1} (p_{4j+2} + p_{4j+3} - p_{4j} - p_{4j+1}) - p_{4k} - p_{4k+1} \leq p_2 \leq r.$$

That means $|L_1 - L_2| \leq r$. Similarly it is easy to show that the conclusion is true for the case of $n = 4k + 2$ and $n = 4k + 3$.

Theorem 3. For any job list $L = \{J_1, J_2, \dots, J_n\}$ with $p_1 \geq p_2 \geq \dots \geq p_n$ and $p_j \in [1, r] (1 \leq r \leq 2)$, algorithm S has worst case performance ratio

$$\frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} \leq \begin{cases} \frac{2+r}{3}, & \frac{3}{2} < r \leq 2 \\ \frac{7}{6}, & \frac{4}{3} \leq r \leq \frac{3}{2} \\ \frac{r+1}{2}, & 1 \leq r < \frac{4}{3} \end{cases} \quad (1)$$

Proof: Suppose (1) is not true. For $\frac{3}{2} \leq r \leq 2$ the following inequalities hold by Lemma 2:

$$\frac{2+r}{3} < \frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} \leq 1 + \frac{r}{2C_{\max}^{OPT}(L)}$$

That means $C_{\max}^{OPT}(L) < \frac{3r}{2(r-1)} < 5$. Similarly $C_{\max}^{OPT}(L) < 5$ also holds for

$\frac{4}{3} \leq r < \frac{3}{2}$. That means there are at most four jobs assigned on any machine in any optimal schedule, i.e., $n \leq 8$. It is easy to prove that algorithm S is optimal if $n < 5$. Now consider $n = 5$. In this case

$$L_1 = p_1 + p_4 + p_5 \geq p_2 + 2 \geq p_2 + p_3 = L_2, \\ C_{\max}^{OPT}(L) \geq p_3 + p_4 + p_5.$$

Hence

$$\frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} \leq \frac{p_1 + p_4 + p_5}{p_3 + p_4 + p_5} \leq \frac{p_1 + (p_4 + p_5)}{p_3 + (p_4 + p_5)} \leq \frac{p_1 + 2}{p_3 + 2} \leq \frac{r + 2}{3}.$$

For the case of $n = 6$, there are exactly three jobs on each machine in any optimal schedule. If $L_1 > L_2$, we get

$$\frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} \leq \frac{p_1 + p_4 + p_5}{p_1 + p_5 + p_6} \leq \frac{p_4 + p_1 + p_5}{p_6 + p_1 + p_5} \leq \frac{r+2}{3}$$

If $L_1 \leq L_2$ we have

$$\frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} \leq \frac{p_2 + p_3 + p_6}{p_1 + p_5 + p_6} \leq \frac{p_2 + (p_3 + p_6)}{p_5 + (p_3 + p_6)} \leq \frac{p_2 + 2}{p_3 + 2} \leq \frac{r+2}{3}.$$

For the case of $n = 7$, the following holds

$$L_2 = p_2 + p_3 + p_6 + p_7 \geq p_2 + p_3 + 2 \geq p_1 + p_2 + p_3 = L_1.$$

In any optimal schedule, exactly four jobs are assigned on one machine and exactly three jobs are assigned on another. If the machine assigned four jobs in optimal schedule has at least one job from set $\{J_1, J_2, J_3\}$, then the following inequality holds:

$$C_{\max}^{OPT}(L) \geq p_3 + p_5 + p_6 + p_7.$$

Hence we get

$$\frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} \leq \frac{p_2 + p_3 + p_6 + p_7}{p_3 + p_5 + p_6 + p_7} \leq \frac{p_2 + 3}{p_5 + 3} \leq \frac{r+3}{4} \leq \frac{r+2}{3}.$$

Otherwise the optimal schedule is that $\{J_1, J_2, J_3\}$ and $\{J_4, J_5, J_6, J_7\}$ are assigned separately on two machines. Let $a = \frac{p_2 + p_3}{2}$, $b = \frac{p_4 + p_5}{2}$, $c = \frac{p_6 + p_7}{2}$. It is easy to see that $a \geq b \geq c$ holds. We analyze the following two cases. In the case of $3a \leq 2b + 2c$, we get

$$\frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} \leq \frac{p_2 + p_3 + p_6 + p_7}{p_4 + p_5 + p_6 + p_7} = \frac{a + c}{b + c} \leq \frac{2b + 5c}{3b + 3c} = \frac{2}{3} + \frac{c}{b + c} \leq \frac{7}{6},$$

the last inequality results from $b \geq c$.

In the case of $3a > 2b + 2c$ we get

$$\frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} \leq \frac{p_2 + p_3 + p_6 + p_7}{p_1 + p_2 + p_3} = \frac{2a + 2c}{3a} = \frac{2}{3} + \frac{2c}{3a} \leq \frac{2}{3} + \frac{2c}{2b + 2c} \leq \frac{7}{6}.$$

For $n = 8$, there are exactly four jobs assigned on each machine and there is a machine on which at least two jobs from $\{J_1, J_2, J_3\}$ are assigned in any optimal schedule. Hence the following inequality holds:

$$C_{\max}^{OPT}(L) \geq \max\{p_2 + p_3 + p_7 + p_8, p_1 + p_6 + p_7 + p_8\}.$$

Therefore if $L_1 \geq L_2$ we get

$$\frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} \leq \frac{p_1 + p_4 + p_5 + p_8}{p_2 + p_3 + p_7 + p_8} \leq \frac{p_1 + p_4 + p_5 + p_8}{p_7 + p_4 + p_5 + p_8} \leq \frac{r+3}{4} \leq \frac{r+2}{3}.$$

If $L_1 \leq L_2$ we get

$$\frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} \leq \frac{p_2 + p_3 + p_6 + p_7}{p_1 + p_6 + p_7 + p_8} \leq \frac{p_2 + p_3 + p_6 + p_7}{p_8 + p_3 + p_6 + p_7} \leq \frac{r+3}{4} \leq \frac{r+2}{3}.$$

By the conclusions above, we get

$$\frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} \leq \max \left\{ \frac{7}{6}, \frac{r+2}{3} \right\} = \begin{cases} \frac{2+r}{3}, & \frac{3}{2} < r \leq 2 \\ \frac{7}{6}, & \frac{4}{3} \leq r \leq \frac{3}{2} \end{cases}$$

Hence (1) is true for $\frac{4}{3} \leq r \leq 2$.

Now we consider the case of $1 \leq r \leq \frac{4}{3}$ according to the four cases of

$n = 4k + 1, 4k + 2, 4k + 3, 4k + 4, k = 0, 1, 2, \dots$. In the following, we will use S_i and S_i^* to denote the job set assigned on machine M_i by algorithm S and optimal algorithm, respectively.

For the case of $n = 4k + 1$, we have $|S_1| = 2k + 1, |S_2| = 2k$, $\max\{|S_1^*|, |S_2^*|\} \geq 2k + 1$. Without loss of generality, suppose $|S_1^*| \geq 2k + 1$. Then it is easy to see that there exists $i \in \{1, 2\}$ satisfying $|S_i^* \cap S_i| \geq k + 1$. If $|S_1^* \cap S_1| \geq k + 1$, then $|S_1 \setminus S_1^*| \leq k$. By

$$L_2 - L_1 = p_2 + p_3 - p_1 + \sum_{j=1}^{k-1} (p_{4j+2} + p_{4j+3} - p_{4j} - p_{4j+1}) - p_{4k} - p_{4k+1} \leq 0,$$

we get $L_1 \geq L_2$. Therefore

$$\begin{aligned} \frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} &= \frac{L_1}{C_{\max}^{OPT}(L)} \leq \frac{\sum_{j \in S_1} p_j}{\sum_{j \in S_1^*} p_j} \leq \frac{\sum_{j \in S_1 \cap S_1^*} p_j + \sum_{j \in S_1 \setminus S_1^*} p_j}{\sum_{j \in S_1 \cap S_1^*} p_j + \sum_{j \in S_1^* \setminus S_1} p_j} \\ &\leq \frac{|S_1 \cap S_1^*| + |S_1 \setminus S_1^*| r}{|S_1 \cap S_1^*| + |S_1^* \setminus S_1|} \\ &\leq \frac{|S_1 \cap S_1^*| + (|S_1| - |S_1 \cap S_1^*|) r}{2k + 1} \\ &\leq \frac{kr + k + 1}{2k + 1} \leq \frac{r + 1}{2}. \end{aligned}$$

If $|S_1^* \cap S_1| \leq k$, then $|S_1^* \cap S_2| \geq k + 1$. By

$$L_1 - L_2 = p_1 + \sum_{j=1}^k (p_{4j} + p_{4j+1} - p_{4j-2} - p_{4j-3}) \leq p_1$$

we get $L_1 \leq L_2 + p_1$. Therefore

$$\begin{aligned} \frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} &\leq \frac{L_2 + p_1}{C_{\max}^{OPT}(L)} \leq \frac{\sum_{j \in S_2} p_j}{\sum_{j \in S_1^*} p_j} \leq \frac{\sum_{j \in S_2 \cap S_1^*} p_j + \sum_{j \in S_2 \setminus S_1^*} p_j + p_1}{\sum_{j \in S_2 \cap S_1^*} p_j + \sum_{j \in S_1^* \setminus S_2} p_j} \\ &\leq \frac{|S_2 \cap S_1^*| + |S_2 \setminus S_1^*| r + r}{|S_2 \cap S_1^*| + |S_1^* \setminus S_2|} \\ &\leq \frac{|S_2 \cap S_1^*| + (|S_2| - |S_2 \cap S_1^*|) r + r}{2k + 1} \\ &\leq \frac{kr + k + 1}{2k + 1} \leq \frac{r + 1}{2}. \end{aligned}$$

For the case of $n = 4k + 2$, we have $|S_1| = 2k + 1$, $|S_2| = 2k + 1$, $\max\{|S_1^*|, |S_2^*|\} \geq 2k + 1$. Without loss of generality, suppose $|S_1^*| \geq 2k + 1$. Therefore there exists $i \in \{1, 2\}$ satisfying $|S_1^* \cap S_i| \geq k + 1$. In the following we consider this case according to the two subcases of $L_1 \geq L_2$ and $L_1 < L_2$.

In this case of $L_1 \geq L_2$, if $|S_1^* \cap S_1| \geq k + 1$ holds, then the following is true:

$$\begin{aligned} \frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} &= \frac{L_1}{C_{\max}^{OPT}(L)} \leq \frac{\sum_{j \in S_1} p_j}{\sum_{j \in S_1^*} p_j} \leq \frac{\sum_{j \in S_1 \cap S_1^*} p_j + \sum_{j \in S_1 \setminus S_1^*} p_j}{\sum_{j \in S_1 \cap S_1^*} p_j + \sum_{j \in S_1^* \setminus S_1} p_j} \\ &\leq \frac{|S_1 \cap S_1^*| + |S_1 \setminus S_1^*| r}{|S_1 \cap S_1^*| + |S_1^* \setminus S_1|} \\ &\leq \frac{|S_1 \cap S_1^*| + (|S_1| - |S_1 \cap S_1^*|) r}{2k + 1} \\ &\leq \frac{kr + k + 1}{2k + 1} \leq \frac{r + 1}{2}. \end{aligned}$$

If $|S_1^* \cap S_1| \leq k$ holds, we consider the following two subcases of $|S_1^*| = 2k + 1$ and $|S_1^*| \geq 2k + 2$.

For the case of $|S_1^*| = 2k + 1$, $|S_2^*| = 2k + 1$ holds by $n = 4k + 2$. By $|S_1| = 2k + 1$ and $|S_1^* \cap S_1| \leq k$ we get $|S_2^* \cap S_1| \geq k + 1$. Therefore

$$\begin{aligned} \frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} &= \frac{L_1}{C_{\max}^{OPT}(L)} \leq \frac{\sum_{j \in S_1} p_j}{\sum_{j \in S_2^*} p_j} \leq \frac{\sum_{j \in S_1 \cap S_2^*} p_j + \sum_{j \in S_1 \setminus S_2^*} p_j}{\sum_{j \in S_1 \cap S_2^*} p_j + \sum_{j \in S_2^* \setminus S_1} p_j} \\ &\leq \frac{|S_1 \cap S_2^*| + |S_1 \setminus S_2^*| r}{|S_1 \cap S_2^*| + |S_2^* \setminus S_1|} \\ &\leq \frac{|S_1 \cap S_2^*| + (|S_1| - |S_1 \cap S_2^*|) r}{2k + 1} \\ &\leq \frac{kr + k + 1}{2k + 1} \leq \frac{r + 1}{2}. \end{aligned}$$

For the case of $|S_1^*| \geq 2k + 2$, by $|S_1^* \cap S_1| \leq k$ we get $|S_1^* \cap S_2| \geq k + 2$. By rules of S algorithm, we have

$$L_1 - L_2 = p_1 + \sum_{j=1}^k (p_{4j} + p_{4j+1} - p_{4j-2} - p_{4j-1}) - p_{4k+2} \leq p_1 - p_{4k+2}.$$

That means $L_1 \leq L_2 + p_1 - p_{4k+2}$. Therefore

$$\begin{aligned} \frac{C_{\max}^S(L)}{C_{\max}^{OPT}(L)} &= \frac{L_1}{C_{\max}^{OPT}(L)} \leq \frac{L_2 + p_1 - p_{4k+2}}{C_{\max}^{OPT}(L)} \\ &\leq \frac{\sum_{j \in S_2 \cap S_1^*} p_j + \sum_{j \in S_2 \setminus S_1^*} p_j + p_1 - p_{4k+2}}{\sum_{j \in S_2 \cap S_1^*} p_j + \sum_{j \in S_1^* \setminus S_2} p_j} \leq \frac{|S_2 \cap S_1^*| + |S_2 \setminus S_1^*| r + r - 1}{|S_2 \cap S_1^*| + |S_1^* \setminus S_2|} \\ &\leq \frac{|S_2 \cap S_1^*| + (|S_2| - |S_2 \cap S_1^*|) r + r - 1}{2k + 2} \leq \frac{kr + k + 1}{2k + 2} \leq \frac{kr + k + 1}{2k + 1} \leq \frac{r + 1}{2}. \end{aligned}$$

Similarly we can prove the case of $L_1 < L_2$.

By the same way used above, we can also show that (1) is true for the case of $n = 4k + 3$ and $n = 4k + 4$ for $1 \leq r \leq \frac{4}{3}$. Now we show the tightness of the bound.

For $\frac{3}{2} < r \leq 2$, Let $L^{(1)} = \{J_1, J_2, J_3, J_4, J_5\}$ with $p_1 = r, p_i = 1, i = 2, \dots, 5$. By the rules of S algorithm, we have $L_1 = r + 2, L_2 = 2$, i.e., $C_{\max}^S(L^{(1)}) = r + 2$. It is easy to see that $C_{\max}^{OPT}(L^{(1)}) = 3$. Hence

$$\frac{C_{\max}^S(L^{(1)})}{C_{\max}^{OPT}(L^{(1)})} = \frac{r+2}{3}.$$

It is easy to show the tightness for $\frac{4}{3} \leq r \leq \frac{3}{2}$ by job list

$$L^{(2)} = \{J_1, J_2, J_3, J_4, J_5, J_6, J_7\} \text{ with } p_1 = p_2 = p_3 = \frac{4}{3}, p_4 = p_5 = p_6 = p_7 = 1$$

and for $1 \leq r \leq \frac{4}{3}$ by job list with $p_1 = p_2 = p_3 = r, p_4 = p_5 = p_6 = p_7 = 1$.

4. Concluding Remarks

In this paper, we consider ordinal on-line scheduling for jobs with known sizes in $[1, r](r \geq 1)$ and non-decreasing processing times on two parallel machines system. Firstly it is proved that the worst case performance ratio of the existing algorithm P_2 can not be improved even if the job processing times are known in $[1, r]$ for any $r \geq 1$. Secondly, an algorithm named S is proposed and its worst case performance ratio is given as follow:

$$R(2, S) = \begin{cases} \frac{2+r}{3}, & \frac{3}{2} < r \leq 2 \\ \frac{7}{6}, & \frac{4}{3} \leq r \leq \frac{3}{2} \\ \frac{r+1}{2}, & 1 \leq r < \frac{4}{3} \end{cases}$$

which is better than algorithm P_2 . Just two machines are considered here. It is an interesting problem to consider general m machines system to design better algorithm.

Acknowledgements

The authors would like to express their thanks to the National Natural Science Foundation of China for financially supporting under Grant No. 11471110 and the Foundation Grant of Education Department of Hunan (No. 16A126).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Graham R.L. (1969) Bounds on Multiprocessing Timing Anomalies. *SIAM Journal on Applied Mathematics*, **17**, 416-429. <https://doi.org/10.1137/0117039>
- [2] Cheng, T.C.E., Kellerer, H. and Kotov, V. (2012) Algorithms Better than LPT for Semi-Online Scheduling with Decreasing Processing Times. *Operations Research Letters*, **40**, 349-352.
- [3] He, Y. and Dósa, G. (2005) Semi-Online Scheduling Jobs with Tightly-Grouped Processing Times on Three Identical Machines. *Discrete Applied Mathematics*, **150**, 140-159.
- [4] He, Y. and Zhang, G. (1999) Semi on-Line Scheduling on Two Identical Machines. *Computing*, **62**, 179-187. <https://doi.org/10.1007/s006070050020>
- [5] Kellerer, H., Kotov, V., Speranza, M.G. and Tuza, Z. (1997) Semi on-Line Algorithms for the Partition Problem. *Operations Research Letters*, **21**, 235-242. [https://doi.org/10.1016/S0167-6377\(98\)00005-4](https://doi.org/10.1016/S0167-6377(98)00005-4)
- [6] Lin, L. and Tan, Z. (2014) Inefficiency of Nash Equilibrium for Scheduling Games with Constrained Jobs: A Parametric Analysis. *Theoretical Computer Science*, **521**, 123-134. <https://doi.org/10.1016/j.tcs.2013.11.012>
- [7] Gupta, S., Dalal, U.D. and Mishra, V.N. (2014) Novel Analytical Approach of Conventional Mapping Scheme with Discrete Hartley Transform in OFDM System. *American Journal of Operations Research*, **4**, 281-292. <https://doi.org/10.4236/ajor.2014.45027>
- [8] Liu, W.P., Sidney, J.B. and Vliet, A.V. (1996) Ordinal Algorithms for Parallel Machine Scheduling. *Operations Research Letters*, **18**, 223-232. [https://doi.org/10.1016/0167-6377\(95\)00058-5](https://doi.org/10.1016/0167-6377(95)00058-5)