Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Online algorithms for scheduling with machine activation cost on two uniform machines*

HAN Shu-guang[1], JIANG Yi-wei[†‡2], HU Jue-liang[2]

([1]Department of Mathematics, Zhejiang University, Hangzhou 310027, China)

([2]Faculty of Science, Zhejiang Sci-Tech University, Hangzhou 310018, China)

[†]E-mail: jywzju@163.com

Received Jan. 28, 2006; revision accepted May 3, 2006

Abstract: In this paper we investigate a variant of the scheduling problem on two uniform machines with speeds 1 and s. For this problem, we are given two potential uniform machines to process a sequence of independent jobs. Machines need to be activated before starting to process, and each machine activated incurs a fixed machine activation cost. No machines are initially activated, and when a job is revealed, the algorithm has the option to activate new machines. The objective is to minimize the sum of the makespan and the machine activation cost. We design optimal online algorithms with competitive ratio of $(2s+1)/(s+1)$ for every $s\geq1$.

Key words: Online algorithm, Competitive analysis, Uniform machine scheduling, Machine activation cost

doi:10.1631/jzus.2007.A0127 **Document code:** A **CLC number:** TP393; O223

INTRODUCTION

In this paper we investigate the uniform machines scheduling problem with machine activation cost. This problem has application in garment production of international trade and is motivated by the following scenario. Import-export company is compared to scheduler in this model, and orders are jobs, which arrive one by one. And garment factories can be regarded as machines. Since jobs should be finished on time, scheduler will choose a reasonable number of machines to make the garments. When the machines accept the orders, a certain amount of cost is needed for the running of machinery and the workers getting familiar with the techniques, etc. The cost is fixed and proportional to the speed of the machine, and occurs with the running of machines.

Formally, the problem considered in this paper can be described as follows. We are given a sequence J of independent jobs with positive processing times (sizes) $J=\{p_1,p_2,\ldots,p_n\}$, which must be non-preemptively scheduled onto two uniform machines (with speeds of 1 and $s\geq1$). We identify jobs with their sizes in this paper. Jobs arrive one by one (online over list) and are to be scheduled irrevocably onto these machines as soon as they are given, without any knowledge of the jobs that will arrive later. There are only two potential uniform machines to process these jobs. If one machine is used to process jobs, it must be activated and the activation cost cannot be neglected. And initially there are no machines activated.

Although the machines are uniform, the costs for activation are different. Moreover, by normalizing all job sizes and machine activation cost, we assume that the activation cost for the machine with speed 1 is 1, and the other with speed s is s, without loss of generality. Let AMC be the total machine activation cost. The load of a machine is the sum of the sizes of the jobs scheduled onto it, and the makespan is the maximum completion time after all jobs are com-

‡ Corresponding author

* Project (No. Y605316) supported by the Natural Science Foundation of Zhejiang Province, China and the Natural Science Foundation of the Education Department of Zhejiang Province (No. 20060578), China

pleted. Then the goal is to minimize the sum of the makespan and the total machine activation cost.

The performance of an online algorithm is measured by its competitive ratio. For a job sequence J and an algorithm A, let $c^A(J)$ (or in short c^A) denote the makespan produced by A and let $c^*(J)$ (or in short c^*) denote the optimal makespan in an offline version. Then the competitive ratio of A is defined as $C=\sup\{c^A(J)/c^*(J)\}$. An online problem has a lower bound ρ if no online deterministic algorithm has a competitive ratio smaller than ρ. An online algorithm is optimal if its competitive ratio matches the lower bound.

This problem is quite different from the classical online uniform machine scheduling problem $Q2|online|C_{\max}$ (Cho and Sahni, 1980), where we typically have two uniform machines, and the scheduler makes no decision regarding the cost of machines that are used to process jobs. There has been a great deal of work on this problem (Epstein *et al.*, 2001; Sgall, 1998). For non-preemptive scheduling on uniform machines the first algorithm with a constant competitive ratio was given by Aspnes *et al.*(1997) and its competitive ratio is 8. This was improved by Berman *et al.*(1997); they designed 5.8285-competitive deterministic and 4.3111-competitive randomized algorithms. Berman *et al.*(1997) also proved lower bounds of 2.4380 for deterministic and 1.8372 for randomized algorithms for non-preemptive scheduling. And the lower bound of 1.8372 has been improved to 2 by Epstein and Sgall (2000). Dessouky *et al.*(1998) considered flowshop scheduling with identical jobs and uniform parallel machines. Noga and Seiden (2001) investigated scheduling problem on two machines with release times. Tan and He (2002) investigated scheduling problem on two identical machines with machine availability constraints.

Imreh and Noga (1999) proposed a variant problem, denoted as $P|online|C_{\max}+m$ and called List Model. The differences are: (1) no machines are initially provided, and there is sufficient large number of identical machines that can be activated (i.e., $m=+\infty$), (2) when a job is revealed the algorithm has the option to activate new machines, and (3) the objective is to minimize the sum of the makespan and the total machine activation cost. Comparing it with our considered problem, we know that the differences are whether $m=2$ and whether the machines are uniform. Hence, we call our problem Restricted List Model on Two Uniform Machines, and denote it as $Q2|online|C_{\max}+AMC$.

Panwalkar and Liman (2002) proposed another offline scheduling problem that there are $m=+\infty$ identical machines which can be activated, and the objective is to find an optimal schedule, the optimal number of machines, and the respective due dates to minimize the weighted sum of earliness, tardiness, and machine activation cost. Cao *et al.*(2005) considered a scheduling problem where finite machines are provided. The objective is to minimize the sum of the total weighted job tardiness penalties and the total machine activation cost.

For the List Model problem, Imreh and Noga (1999) presented an online algorithm A_ρ with a competitive ratio of at most $(1+\sqrt{5})/2\approx1.618$, while the lower bound was 4/3. Dósa and He (2004) made an improvement by presenting an algorithm with a competitive ratio of at most $(3+2\sqrt{6})/5\approx1.5798$. Jiang and He (2005) extended it to consider preemptive online and semi-online algorithms for List Model problem. It is remarkable that the good performance of the algorithms in (Imreh and Noga, 1999; Dósa and He, 2004; Jiang and He, 2005; 2006) is based on that we are allowed to activate a large number of machines if needed. He *et al.*(2006) extended it to consider online algorithms for List Model problem with finite identical machines. To our best knowledge, there is no result about $Q2|online|C_{\max}+AMC$.

In this paper, we consider the problem $Q2|online|C_{\max}+AMC$ and design optimal online algorithms with competitive ratio of $(2s+1)/(s+1)$ for every $s\geq1$. The paper is organized as follows. In Section 2 we present some notations and preliminary results. Then we consider the problem with $1\leq s\leq\phi$ and $s>\phi$ (where $\phi=(1+\sqrt{5})/2\approx1.618$ is the golden ratio) in Sections 3 and 4, respectively. Some remarks are presented in Section 5.

PRELIMINARY KNOWLEDGE

In the remainder of the paper, we use the following notations to simplify the presentation. Denote $p_j^{\max}=\max\{p_i|i=1,\cdots,j\}$ and $P_j=\sum_{i=1}^{j}p_i$. Let M_1

and M_s be the two potential uniform machines with speeds 1 and s, respectively. We call moment j as the time right after the jth job is scheduled. Let $L_{i,j}$ denote the current load of machine M_i at moment j (j>0) in an online algorithm A, i=1 or s. Let C_j^A be the current makespan yielded by Algorithm A at moment j, and m_j be the current machine activation cost yielded by Algorithm A at moment j. Denote by $c_j^A = C_j^A + m_j$ the objective function value produced by Algorithm A at moment j. Let C^* and m^* be the makespan and the machine activation cost in an optimal solution of the offline version, respectively. Then the objective function value yielded by Algorithm A and the optimal value of offline version are $c^A = C_n^A + m_n$ and $c^* = C^* + m^*$, respectively.

Now we present the lower bound of the considered problem.

Theorem 1 There is no online algorithm for the problem $Q2|online|C_{\max}+AMC$ with a competitive ratio of smaller than $(2s+1)/(s+1)$.

Proof Assume that an algorithm A exists and it has a competitive ratio $C<(2s+1)/(s+1)$.

(1) We first show $C\geq(2s+1)/(s+1)$ for the case of $1\leq s\leq\phi$. The first two jobs $p_1=N$ and $p_2=sN$ arrive, where N is a very large positive number. We can conclude that both machines must be activated right after scheduling the first two jobs. Otherwise, no more jobs arrive.

We assume only one machine is activated by Algorithm A. It implies that

$$c^A \geq \min\{1+N+sN, s+(N+sN)/s\} = s+N(1+s)/s,$$

which, together with $c^*=N+s+1$, leads to

$$\frac{c^A}{c^*} \geq \frac{s+N(1+s)/s}{N+s+1} \to \frac{s+1}{s} \geq \frac{2s+1}{s+1}.$$

The above formula holds because we can choose N to be large enough and with the assumption of $s\leq\phi$.

Hence, Algorithm A must activate the two machines to process the first two jobs. Then how to choose the following jobs to avoid $C<(2s+1)/(s+1)$?

Note that the lower bound $(2s+1)/(s+1)$ of $Q2|online|C_{\max}$ is obtained by using the adversary method, where an adversary presents the online algorithm with several different sequences that make the algorithm unable to work well simultaneously. Thus, for $Q2|online|C_{\max}+AMC$, we just choose these sequences as the following jobs after activating the two machines, but all the job sizes of these sequences are multiplied by a sufficiently large positive number L such that the sizes of the first two jobs can be ignored when $L\to+\infty$. Hence, the arguments for obtaining the lower bound of $Q2|online|C_{\max}$ can work for our problem. It yields that the lower bound of our problem is at least $(2s+1)/(s+1)$ when $1\leq s\leq\phi$.

(2) We next show $C\geq(2s+1)/(s+1)$ for the case $s>\phi$. Consider a sequence of jobs with each job $p_i=\varepsilon$ $\forall i$, where ε is a very small positive number. If p_1 is assigned to machine M_s, then no other new jobs arrive. Therefore we have $c^A=s+\varepsilon/s$ while the optimal value is $1+\varepsilon$. It follows that $C\geq(s+\varepsilon/s)/(1+\varepsilon)\to s>(2s+1)/(s+1)$ because we can choose ε to be arbitrary small and $s>\phi$. So we assume that Algorithm A assigned the first job to machine M_1.

Moreover, we can claim that Algorithm A must assign the first k jobs (if any) to M_1 to avoid $C<(2s+1)/(s+1)$, with k satisfying $P_k=(s^3-s)/(s^2-s-1)$. Otherwise, we let p_l ($1<l\leq k$) be the first job to be assigned to machine M_s, and no other new jobs arrive after scheduling p_l. Then the total size of jobs assigned to M_1 is P_{l-1}. And we have

$$c^A = 1+s+P_{l-1} = 1+s+P_l-p_l = 1+s+P_l-\varepsilon.$$

If $P_l\leq s$, then the optimal value is $1+P_l$. It follows that

$$C \geq \frac{1+s+P_l-\varepsilon}{1+P_l} \geq \frac{2s+1-\varepsilon}{s+1} \to \frac{2s+1}{s+1} \quad (\varepsilon\to 0).$$

If $s<P_l\leq P_k=(s^3-s)/(s^2-s-1)$, then the optimal cost is at most $s+P_l/s$. Similarly, we can obtain that

$$C \geq \frac{1+s+P_l-\varepsilon}{s+P_l/s} \geq \frac{2s+1-\varepsilon}{s+1} \to \frac{2s+1}{s+1} \quad (\varepsilon\to 0).$$

Thus Algorithm A must assign the first k jobs to the machine M_1 completely. Then no other new jobs arrive. It implies that the objective value yielded by Algorithm A is $1+P_k$, while the optimal cost is at most $s+P_k/s$. It follows that $C\geq(1+P_k)/(s+P_k/s)=(2s+1)/(s+1)$ due to the value of P_k. Hence the desired lower bound is obtained and the proof of Theorem 1 is completed.

The following lemma gives a lower bound of the optimal value.

Lemma 1 The optimal value of the considered problem satisfies

$$c^*\geq\min\left\{1+P_n, s+\frac{P_n}{s}, \max\left(1+s+\frac{P_n}{s+1}, 1+s+\frac{p_n^{\max}}{s}\right)\right\}.$$

In particular, $c^*\geq s+1$ when $P_n\geq s$.

Proof If only one machine is activated in an optimal solution, then we have $c^*\geq\min\{1+P_n, s+P_n/s\}$. Otherwise, the optimal makespan is at least $\max\{P_n/(s+1), p_n^{\max}/s\}$, which follows that

$$c^*\geq\max\{1+s+P_n/(s+1), 1+s+p_n^{\max}/s\}.$$

The following lemma is easy to obtain.

Lemma 2 Let x and y be two positive numbers. Let $f_1=(1+s+x/s)/(1+x+y)$ and $f_2=(1+s+x/s)/[s+(x+y)/s]$.

(1) If $x+y>2s$, $x\geq sy$ and $x\geq s$, then $\max\{f_1, f_2\}\leq(2s+1)/(s+1)$;

(2) If $x+y>s$ and $x\geq sy$, then $\max\{f_1, f_2\}\leq(2s+1)/(s+1)$.

AN OPTIMAL ONLINE ALGORITHM FOR $1\leq s\leq\phi$

In this section, we design an optimal online algorithm for $1\leq s\leq\phi=(1+\sqrt{5})/2\approx1.618$, which can be formally described as follows:

Algorithm H1

Step 1: If $p_1<s$, activate M_1, and schedule p_1 onto M_1. Otherwise, activate M_s, and schedule p_1 onto M_s. Let $k=1$.

Step 2: If no new job arrives, stop. Otherwise, let $k=k+1$.

Step 3: If only M_1 is activated, and

Step 3.1: If $L_{1,k-1}+p_k<s$, schedule p_k onto M_1. Return to Step 2.

Step 3.2: If $L_{1,k-1}+p_k\geq s$, activate M_s and schedule p_k onto M_s. Return to Step 2.

Step 4: If only M_s is activated, and

Step 4.1: If $L_{s,k-1}+p_k<2s$, schedule p_k onto M_s. Return to Step 2.

Step 4.2: If $L_{s,k-1}+p_k\geq 2s$, activate M_1 and schedule p_k onto M_1. Return to Step 2.

Step 5: If both machines are activated, schedule p_k by Post-Greedy rule (Post-Greedy rule means that schedule p_k onto some machine such that the job is completed as early as possible. That is, if $p_k+L_{1,k-1}\leq(p_k+L_{s,k-1})/s$, p_k is scheduled onto M_1, and onto M_s otherwise). Return to Step 2.

Lemma 3 If only one machine is activated by Algorithm H1, then $c^{\mathrm{H1}}/c^*\leq(2s+1)/(s+1)$.

Proof If only machine M_1 is activated, then $P_n<s$ from the algorithm rule. It is clear that $c^{\mathrm{H1}}=1+P_n$, while from Lemma 1, $c^*\geq1+P_n=c^{\mathrm{H1}}$ holds trivially for $P_n<s$. Obviously Algorithm H1 is optimal.

If only machine M_s is activated, furthermore if only one job arrives, then it is obvious that the current scheduling is optimal. Otherwise, there are at least two jobs revealed, then we have $p_1\geq s$ and $s\leq P_n<2s$ by the algorithm rule. Thus $c^{\mathrm{H1}}=s+P_n/s<s+(2s)/s=s+2$, while from Lemma 1, we get $c^{\mathrm{H1}}/c^*\leq(2+s)/(s+1)=1+(s+1)^{-1}\leq(2s+1)/(s+1)$. Now the proof of Lemma 3 is completed.

We next focus on the cases that both machines are activated. Let p_l be the job that determines the makespan yielded by Algorithm H1.

Lemma 4 If p_l is scheduled by Step 1, i.e., $l=1$, then $c^{\mathrm{H1}}/c^*\leq(2s+1)/(s+1)$.

Proof If p_1 is scheduled on M_1, i.e. $p_1<s$, then we have $c^{\mathrm{H1}}=1+s+p_1<1+s+s=2s+1$. It is easy to obtain that $P_n>s$ because the machine M_s is also activated after scheduling p_1. Therefore, by Lemma 1, we have $c^*\geq1+s$. It follows that $c^{\mathrm{H1}}/c^*\leq(2s+1)/(s+1)$.

If p_1 is scheduled on M_s, i.e. $p_1\geq s$, then we can conclude that p_1 is the unique job processed on M_s, and we have $c^{\mathrm{H1}}=1+s+p_1/s$ and $p_1/s\geq L_{1,n}$, implying $p_l=p_n^{\max}$. It is easy to obtain that $p_1+L_{1,n}=P_n>2s$ because the machine M_1 is also activated after assigning p_1. From Lemma 1, we get $c^*\geq\min\{1+p_1+L_{1,n}, s+(p_1+L_{1,n})/s, 1+s+p_1/s\}$. Therefore, by Lemma 2(1) with $x=p_1$ and $y=L_{1,n}$, we have

$$\frac{c^{\mathrm{H1}}}{c^*}\leq\max\left\{\frac{1+s+p_1/s}{1+p_1+L_{1,n}}, \frac{1+s+p_1/s}{s+(p_1+L_{1,n})/s}, 1\right\}\leq\frac{2s+1}{s+1}.$$

The proof is completed.

Lemma 5 If p_l is scheduled by Step 3, then $c^{\mathrm{H1}}/c^*\leq(2s+1)/(s+1)$.

Proof p_l is scheduled by Step 3, we conclude that the two machines are activated in the order of M_1, M_s, which implies that $P_n \geq s$. Two cases are considered according to the assignment of p_l.

Case 1 p_l is scheduled onto machine M_1 by Step 3.1. Then we get $C_n^{\mathrm{H1}} = L_{1,l-1} + p_l < s$, resulting in $c^{\mathrm{H1}} \leq 1+s+s=1+2s$. While according to the rule of Algorithm H1 and Lemma 1, we have $c^* \geq 1+s$ and thus $c^{\mathrm{H1}}/c^* \leq (2s+1)/(s+1)$.

Case 2 p_l is scheduled onto machine M_s by Step 3.2. It is clear that p_l is the unique job scheduled onto machine M_s, which yields that $c^{\mathrm{H1}}=1+s+p_l/s$. Together with the definition of p_l, we have $p_l = p_n^{\max}$. While from Lemma 1, it follows that $c^* \geq \min\{1+p_l+L_{1,n}, s+(p_l+L_{1,n})/s, 1+s+p_l/s\}$.

It is obvious that $p_l+L_{1,n}=P_n \geq s$ and $p_l/s \geq L_{1,n}$ due to the definition of p_l. Therefore, by Lemma 2(2) with $x=p_l$ and $y=L_{1,n}$, we have

$$\frac{c^{\mathrm{H1}}}{c^*} \leq \max\left\{\frac{1+s+p_l/s}{1+p_l+L_{1,n}}, \frac{1+s+p_l/s}{s+(p_l+L_{1,n})/s}, 1\right\} \leq \frac{2s+1}{s+1}.$$

Therefore, the proof of Lemma 5 is completed.

Lemma 6 If p_l is scheduled by Step 4, then $c^{\mathrm{H1}}/c^* \leq (2s+1)/(s+1)$.

Proof Since p_l is scheduled by Step 4, we can conclude that the two machines are activated in the order of M_s, M_1, which implies that $P_n \geq 2s$. Two cases are considered according to the assignment of p_l by Algorithm H1.

Case 1 p_l is scheduled onto machine M_s by Step 4.1, then we have $L_{s,l-1}+p_l<2s$. It follows that $c^{\mathrm{H1}}=1+s+(L_{s,l-1}+p_l)/s<1+s+2=3+s$. It is easy to obtain that $c^* \geq 2+s$ from $P_n \geq 2s$ and Lemma 1. Hence, together with $1 \leq s$, we have $c^{\mathrm{H1}}/c^* \leq (3+s)/(2+s)=1+(2+s)^{-1} \leq (2s+1)/(s+1)$.

Case 2 p_l is scheduled onto machine M_1 by Step 4.2. It is clear that p_l is the unique job scheduled onto machine M_1 and $p_l>L_{s,n}/s$ due to the definition of p_l. We have $c^{\mathrm{H1}}=1+s+p_l$.

If only one machine is activated in the optimal solution, then with $P_n \geq 2s$ we have $c^* \geq \min\{1+P_n, s+P_n/s\}=s+P_n/s$. It is true that $L_{s,n}>p_l>s$ by the rule of Step 1. Together with $1 \leq s \leq \phi$, we obtain

$$\frac{c^{\mathrm{H1}}}{c^*} \leq \frac{1+s+p_l}{s+P_n/s} = \frac{1+s+p_l}{s+(p_l+L_{s,n})/s} \leq \frac{1+s+p_l}{s+(p_l+s)/s} = \frac{1+s+p_l}{1+s+p_l/s} \leq s \leq \frac{2s+1}{s+1}.$$

Otherwise, both machines are activated in the optimal solution, and then we have

$$c^* \geq \max\{1+s+P_n/(s+1), 1+s+p_n^{\max}/s\}.$$

If $p_l>sL_{s,n}$ (implying $p_l = p_n^{\max}$), then we have $c^* \geq 1+s+p_l/s$, which leads to

$$\frac{c^{\mathrm{H1}}}{c^*} \leq \frac{1+s+p_l}{1+s+p_l/s} \leq s \leq \frac{2s+1}{s+1},$$

together with $1 \leq s \leq \phi$. If $p_l \leq sL_{s,n}$, then we obtain $c^* \geq 1+s+(p_l+L_{s,n})/(s+1)$ with $P_n=p_l+L_{s,n}$. Hence, we have

$$\frac{c^{\mathrm{H1}}}{c^*} \leq \frac{1+s+p_l}{1+s+\frac{p_l+L_{s,n}}{s+1}} \leq \frac{1+s+sL_{s,n}}{1+s+L_{s,n}} \leq s \leq \frac{2s+1}{s+1}.$$

Lemma 7 If p_l is scheduled by Step 5, then $c^{\mathrm{H1}}/c^* \leq (2s+1)/(s+1)$.

Proof p_l is scheduled by Step 5 of Algorithm H1, then we have $P_n>s$. We distinguish two cases according to the number of machines activated in the optimal solution.

Case 1 Only one machine is activated in the optimal solution, and then we have

$$c^* \geq \min\{1+P_n, s+P_n/s\}=s+P_n/s \text{ due to } P_n>s.$$

We claim that $c^{\mathrm{H1}} \leq 1+s+P_n/s$ by the following reason. If p_l is scheduled onto M_s, it is obvious that $C_n^{\mathrm{H1}}=(p_l+L_{s,l-1})/s \leq P_n/s$. Otherwise, by the Post-Greedy rule of Algorithm H1, we have $C_n^{\mathrm{H1}}=p_l+L_{1,l-1} <(p_l+L_{s,l-1})/s \leq P_n/s$. Hence, we have

$$\frac{c^{\mathrm{H1}}}{c^*} \leq \frac{1+s+P_n/s}{s+P_n/s} = 1+\frac{1}{s+P_n/s} < 1+\frac{1}{s+1} \leq 1+\frac{s}{s+1} \leq \frac{2s+1}{s+1}.$$

Case 2 Both machines are activated in the optimal solution, and then the optimal machine activation cost is $1+s$, which is the same as the machine activation cost of Algorithm H1. In order to obtain $c^{\mathrm{H1}}/c^*=(1+s+C_n^{\mathrm{H1}})/(1+s+C^*) \leq (2s+1)/(s+1)$, we only need to prove that $C_n^{\mathrm{H1}}/c^* \leq (2s+1)/(s+1)$ in the following argument.

It is easy to obtain that $P_n \le (1+s)C^*$ and $p_l \le p_n^{\max} \le sC^*$.

If p_l is assigned to M_1, we have $C_n^{\text{H1}} = L_{1,l-1} + p_l \le (L_{s,l-1}+p_l)/s$ by Post-Greedy rule of Algorithm H1. It yields that $s(L_{s,l-1}+p_l) \le L_{s,l-1}+p_l$, which follows that

$$\begin{aligned}(s+1)C_n^{\text{H1}} &= (s+1)(L_{1,l-1}+p_l) = s(L_{1,l-1}+p_l)+(L_{1,l-1}+p_l)\\ &\le L_{s,l-1}+p_l+L_{1,l-1}+p_l \le P_n+p_l\\ &\le (s+1)C^*+sC^*=(2s+1)C^*.\end{aligned}$$

Otherwise, if p_l is assigned to M_s, then we have $C_n^{\text{H1}} = (p_l+L_{s,l-1})/s \le L_{1,l-1}+p_l$ by Post-Greedy rule of Algorithm H1. Thus

$$\begin{aligned}(s+1)C_n^{\text{H1}} &\le L_{s,l-1}+p_l+L_{1,l-1}+p_l \le P_n+p_l\\ &\le (s+1)C^*+sC^*=(2s+1)C^*.\end{aligned}$$

Therefore, the proof of Lemma 7 is completed.

Theorem 2 The competitive ratio of Algorithm H1 is $C \le (2s+1)/(s+1)$, when $1 \le s \le \phi$, and it is optimal.

Proof The result is direct from Lemmas 3~7, and Theorem 1 shows that Algorithm H1 is optimal.

Note that our result exactly matches that on two identical machines scheduling with machine activation cost when $s=1$ (He *et al*., 2006).

AN OPTIMAL ONLINE ALGORITHM FOR $s>\phi$

In this section, we present an optimal online algorithm for the problem with $s>\phi$, which can be formally described as follows:

Algorithm H2

Step 1: If $p_1>s$, activate M_s and schedule p_1 and all the future jobs onto M_s, stop. Otherwise, activate M_1, and schedule p_1 onto M_1. Let $k=1$.

Step 2: If no new job arrives, stop. Otherwise, let $k=k+1$.

Step 3: If only M_1 is activated and $L_{1,k-1}+p_k<s$, schedule p_k onto M_1. Otherwise, activate M_s and schedule p_k onto M_s. Return to Step 2.

Step 4: If both machines are activated, schedule p_k by Post-Greedy rule. Return to Step 2.

Remark 1 If $p_1>s$, then Algorithm H2 only activates the machine M_s and schedules all the jobs on it. Although in this case we can also activate the machine M_1 to share a part of the load of machine M_s, it does not help to reduce the competitive ratio of Algorithm H2.

Lemma 8 If only one machine is activated by Algorithm H2, then $c^{\text{H2}}/c^* \le (2s+1)/(s+1)$.

Proof If Algorithm H2 only activates M_1, then we have $P_n<s$ by the rule of Step 3. It is obvious that Algorithm H2 is optimal. Otherwise, H2 only activates M_s, then we have $P_n \ge p_1>s$ and $c^{\text{H2}}=s+P_n/s$ by the rule of Step 1. By Lemma 1 and $1+P_n>s+P_n/s$ due to $P_n>s$, we have

$$\begin{aligned}c^* &\ge \min\{1+P_n,\ s+P_n/s,\ 1+s+P_n/(s+1)\}\\ &= \min\{s+P_n/s,\ 1+s+P_n/(s+1)\}.\end{aligned}$$

Then together with $s>(1+\sqrt{5})/2$, we get

$$\begin{aligned}\frac{c^{\text{H2}}}{c^*} &\le \max\left\{1, \frac{s+P_n/s}{1+s+P_n/(s+1)}\right\}\\ &< \max\left\{1, \frac{s+P_n/s}{s+P_n/(s+1)}\right\} \le \max\left\{1, \frac{P_n/s}{P_n/(s+1)}\right\}\\ &=1+1/s \le (2s+1)/(s+1).\end{aligned}$$

Therefore, the proof of Lemma 8 is completed.

We next consider the cases that both machines are activated by Algorithm H2. Let p_l be the job that determines the makespan yielded by Algorithm H2.

Lemma 9 If p_l is scheduled by Step 1, then $c^{\text{H2}}/c^* \le (2s+1)/(s+1)$.

Proof The proof is similar to that of Lemma 4, so we omit it here.

Lemma 10 If p_l is scheduled by Step 3, then $c^{\text{H2}}/c^* \le (2s+1)/(s+1)$.

Proof Since both machines are activated, we have $P_n>s$ due to the rule of Step 3. Two following cases are considered according to the position of p_l.

Case 1 p_l is scheduled onto M_1, then from the algorithm rule, we have

$$c^{\text{H2}}=1+s+p_l+L_{1,l-1}<1+s+s=2s+1.$$

By $P_n>s$ and Lemma 1, we have $c^* \ge s+1$. It follows that $c^{\text{H2}}/c^* \le (2s+1)/(s+1)$.

Case 2 p_l is scheduled onto M_s, then the proof is similar to that of Case 2 in Lemma 5, so we omit it.

Therefore, the proof of Lemma 10 is completed.

Lemma 11 If p_l is scheduled by Step 4, then $c^{H2}/c^{*} \leq(2s+1)/(s+1)$.

Proof The proof is similar to that of Lemma 7, so we omit it.

Theorem 3 The competitive ratio of Algorithm H2 is $C\leq(2s+1)/(s+1)$, when $s>\phi=(1+\sqrt{5})/2$, and it is optimal.

Proof The competitive ratio of Algorithm H2 is obtained directly from Lemmas 8~11. And the optimality of Algorithm H2 is a direct consequence of Theorem 1.

CONCLUSION

In this paper, we considered the problem of $Q2|online|C_{\max}+AMC$. We showed that, due to the machine activation cost, the considered problem becomes harder to approximate than the classical scheduling problem on two uniform machines $Q2|online|C_{\max}$ with regard to competitive analysis. We designed optimal online algorithms with competitive ratios of $(2s+1)/(s+1)$ for all values of s. And each algorithm consists of two parts, an activating strategy, which decides when a potential machine is activated, and a scheduling algorithm, which assigns jobs to machines.

For future research, it is of interest to develop algorithms for the general Restricted List Model on Uniform Machines problem $Qm|online|C_{\max}+AMC$ for $m\geq3$.

References

Aspnes, J., Azar, Y., Fiat, A., Plotkin, S., Waarts, O., 1997. On-line load balancing with applications to machine scheduling and virtual circuit routing. *Journal of the ACM*, **44**(3):486-504. [doi:10.1145/258128.258201]

Berman, P., Charikar, M., Karpinski, M., 1997. On-line Load Balancing for Related Machines. Proceedings of the 5th Workshop on Algorithms and Data Structures. Springer-Verlag, Berlin, **1272**:116-125.

Cao, D., Chen, M., Wan, G., 2005. Parallel machine selection and job scheduling to minimize machine cost and job tardiness. *Computer & Operations Research*, **32**(8):1995-2012. [doi:10.1016/j.cor.2004.01.001]

Cho, Y., Sahni, S., 1980. Bounds for list schedules on uniform processors. *SIAM Journal on Computing*, **9**(1):91-103. [doi:10.1137/0209007]

Dessouky, M.M., Dessouky, M.I., Verma, S., 1998. Flowshop scheduling with identical jobs and uniform parallel machines. *European Journal of Operational Research*, **109**(3):620-631. [doi:10.1016/S0377-2217(97)00194-X]

Dósa, G., He, Y., 2004. Better online algorithms for scheduling with machine cost. *SIAM Journal on Computing*, **33**(5): 1035-1051. [doi:10.1137/S009753970343395X]

Epstein, L., Noga, J., Seiden, S., Sgall, J., Woeginger, G.J., 2001. Randomized on-line scheduling on two uniform machines. *Journal of Scheduling*, **4**(2):71-92. [doi:10.1002/jos.60]

Epstein, L., Sgall, J., 2000. A lower bound for on-line scheduling on uniformly related machines. *Operations Research Letters*, **26**(1):17-22. [doi:10.1016/S0167-6377(99)00062-0]

He, Y., Han, S.G., Jiang, Y.W., 2006. Online algorithms for scheduling with machine activation cost. *Asia-Pacific Journal of Operations research*, in press.

Imreh, C., Noga, J., 1999. Scheduling with Machine Cost. Proc. RANDOM APPROX'99. *Lecture Notes in Computer Science*, **1671**:168-176.

Jiang, Y.W., He, Y., 2005. Preemptive online algorithms for scheduling with machine cost. *Acta Informatica*, **41**(6): 315-340. [doi:10.1007/s00236-004-0156-9]

Jiang, Y.W., He, Y., 2006. Semi-online algorithms for scheduling with machine cost. *Journal of Computer Science & Technology*, **21**(6):984-988.

Noga, J., Seiden, S.S., 2001. An optimal online algorithm for scheduling two machines with release times. *Theoretical Computer Science*, **268**(1):133-143. [doi:10.1016/S0304-3975(00)00264-4]

Panwalkar, S., Liman, S.D., 2002. Single operation earliness-tardiness scheduling with machine activation cost. *IIE Transactions*, **34**(5):509-513. [doi:10.1023/A:1013539825022]

Sgall, J., 1998. On-line Scheduling. *In*: Fiat, A., Woeginger, G.J. (Eds.), Online Algorithms: The State of the Art. Springer-Verlag, Berlin, **1442**:196-231. [doi:10.1007/BFb0029570]

Tan, Z., He, Y., 2002. Optimal online algorithm for scheduling on two identical machines with machine availability constraints. *Information Processing Letters*, **83**(6):323-329. [doi:10.1016/S0020-0190(02)00211-9]