PROGRESS IN NATURAL SCIENCE Vol. 17, No. 11, November 2007

RESEARCH PAPERS

Online uniform machine covering with the known largest size*

Cao Shunjuan[1,2] and Tan Zhiyi[1,3**]

(1. Department of Mathematics, Zhejiang University, Hangzhou 310027, China; 2. Department of Mathematics, Zhejiang Forestry University, Hangzhou 311300, China; 3. State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou 310027, China)

Accepted on June 3, 2007

Abstract This paper investigates the semi-online scheduling problem with the known largest size on two uniform machines. The objective is to maximize the minimum machine completion time. Both lower bounds and algorithms are given. Algorithms are optimal for the majority values of $s \geqslant 1$, where s is the speed ratio of the two machines. The largest gap between the competitive ratio and the lower bound is about 0.064. Moreover, the overall competitive ratio 2 matches the overall lower bound.

Keywords: scheduling and covering, uniform machine, design and analysis of algorithm, online, competitive ratio.

This paper considers the semi-online machine covering problem on two uniform machines with the known largest size. Machine covering problem has application in the sequencing of maintenance actions for modular gas turbine aircraft engines[1]. New applications in online bandwidth allocation and resource allocation were reported recently[2]. The problem discussed in this paper can be described as follows. We are given with a sequence $J_1, J_2, \cdots, J_n$ of independent jobs, each job J_j with a positive size p_j. The largest size of all jobs $p_{\max} = \max_{1 \leqslant j \leqslant n} p_j$ is known in advance. W.l.o.g., we assume $p_{\max} = 1$. Jobs arrive one by one, and we are required scheduling jobs irrevocably on machines as soon as they are given, without any knowledge of the successive jobs except that they have the size less than $p_{\max}$. Let M_1, M_2 be two parallel machines. The speed of M_i is s_i, $i=1, 2$, i.e., the time used for J_j to be scheduled on M_i is p_j / s_i, $j=1, 2, \cdots, n$, $i=1, 2$. Assume $1 = s_1 \leqslant s_2 < \infty$, and let $s = s_2 / s_1$ be the speed ratio of the two machines. Jobs and machines are available at time zero, and no preemption is allowed. The goal is to maximize the minimum machine completion time. We denote the problem by $Q2|\max|C_{\min}$.

Scheduling problems with partial information of future jobs are called semi-online problems[3]. Algorithms for semi-online problems are called semi-online algorithms. The quality of the performance of a semi-online algorithm is measured by its competitive ratio. We define the competitive ratio for maximization problems. For an instance I and an algorithm A, let $C^A(I)$ (or shortly C^A) be the objective value produced by A and let $C^*(I)$ (or shortly C^*) be the optimal value in an offline version. Then the competitive ratio of A is defined as the smallest number c such that for any I, $C^*(I) \leqslant c C^A(I)$. A semi-online scheduling problem has a lower bound ρ if there is no semi-online algorithm with a competitive ratio smaller than ρ. A semi-online algorithm A is called optimal if its competitive ratio matches the lower bound of the problem.

Different kinds of partial information give rise to different semi-online problems, such as known total size[3] (denoted by sum), known the largest size[4] (denoted by max), known the optimal value[5] (denoted by opt), and etc. Among these problems, that in which the largest size is known in advance seems to be the most difficult for algorithm design and analysis. For example, there are semi-online algorithms for $Pm|\text{sum}|C_{\max}$ or $Pm|\text{opt}|C_{\max}$ with competitive ratio smaller than that of $Pm|\text{online}|C_{\max}$[6,5]. But no such algorithm has been reported for $Pm|\max|C_{\max}$ to the authors' knowledge. Semi-online algorithm for $Q2|\text{sum}|C_{\min}$ or $Q2|\text{opt}|C_{\min}$ is optimal for any $s \geqslant$

* Supported by National Natural Science Foundation of China (Grant Nos. 10671177 and 60021201) and the Natural Science Foundation of Zhejiang Forestry University (Grant No. 2006FK36)

** To whom correspondence should be addressed. E-mail: tanzy@zju.edu.cn

1[7,2]. However, known algorithm for $Q2|\max|C_{\min}$ is only optimal for $s=1$ and $\frac{1+\sqrt{5}}{2}$, and the largest gap between the competitive ratio and the lower bound is about 0.55[8].

In this paper, we will present the improved lower bounds and semi-online algorithms for $Q2|\max|C_{\min}$. In Section 1, we prove the lower bound of the problem is at least

$$\begin{cases} \frac{s+2}{s+1} & 1\leqslant s\leqslant\sqrt{2} \\ s & \sqrt{2}< s\leqslant\frac{1+\sqrt{5}}{2} \\ \frac{s+1}{s} & \frac{1+\sqrt{5}}{2}< s\leqslant 1+\sqrt{2} \\ \frac{s^2+s+1+\sqrt{s^4-s^2+2s+1}}{s^2+2s} & s\geqslant 1+\sqrt{2} \end{cases}$$

In Section 2, we present semi-online algorithms with competitive ratio

$$\begin{cases} \frac{s+2}{s+1} & 1\leqslant s\leqslant\sqrt{2} \\ s & \sqrt{2}< s\leqslant\frac{1+\sqrt{5}}{2} \\ \frac{s+1}{s} & \frac{1+\sqrt{5}}{2}< s\leqslant s_1 \\ \frac{s+1+\sqrt{5s^2+6s+1}}{2(s+1)} & s_1< s\leqslant s_2 \\ \frac{s^2+s+1+\sqrt{s^4-s^2+2s+1}}{s(s+2)} & s> s_2 \end{cases}$$

where

$$s_1=\frac{1}{3}+\frac{1}{3}\left(\frac{47}{2}-\frac{3\sqrt{93}}{2}\right)^{\frac{1}{3}}+\frac{1}{3}\left(\frac{47}{2}+\frac{3\sqrt{93}}{2}\right)^{\frac{1}{3}}\approx 2.148$$

and

$$s_2=\frac{2}{3}+\frac{1}{3}(116-6\sqrt{78})^{\frac{1}{3}}+\frac{1}{3}(116+6\sqrt{78})^{\frac{1}{3}}\approx 3.836$$

Hence, the algorithms are optimal for $s\in[1, 2.148)\cup[3.836, \infty)$. The largest gap between the competitive ratio and the lower bound is about 0.064. When s tends to ∞, both the competitive ratio and the lower bound tend to 2, which is also the overall competitive ratio and lower bound of the problem.

1 Lower bounds

The lower bounds of $Q2|\max|C_{\min}$ will be proved through Lemmas 1—3.

Lemma 1. The competitive ratio of any semi-online algorithm A for problem $Q2|\max|C_{\min}$ is at least $\frac{s+2}{s+1}$ when $1\leqslant s\leqslant\sqrt{2}$.

Proof. Let the first job J_1 be the largest job with size 1. We consider two cases.

Case 1. J_1 is assigned to M_1. The sequence continues with a job J_2 of size $\frac{1}{s+1}$. If J_2 is also assigned to M_1, then no job comes. We get $C^A=0$, while $C^*=\frac{1}{s+1}$. It follows that $\frac{C^*}{C^A}=\infty>\frac{s+2}{s+1}$. If J_2 is assigned to M_2, then the third job J_3 of size $\frac{s^2+s-1}{s+1}$ comes. If J_3 is assigned to M_1, we have $C^A=\frac{1}{s(s+1)}$, and $C^*=1$ by assigning J_1 to M_1 and the other two jobs to M_2. It follows that $\frac{C^*}{C^A}=s(s+1)>\frac{s+2}{s+1}$. Otherwise, if we assign J_3 to M_2, then the last job J_4 with size 1 comes. We have $C^A=1$, and $C^*=\frac{s+2}{s+1}$ by assigning J_1, J_2 to M_1 and J_3, J_4 to M_2. It implies that $\frac{C^*}{C^A}=\frac{s+2}{s+1}$.

Case 2. J_1 is assigned to M_2. The sequence continues with a job J_2 of size $\frac{1+s-s^2}{s^2+s}$. If J_2 is also assigned to M_2, then no job comes. So we get $C^A=0$, and $C^*=\frac{1+s-s^2}{s^2+s}$, then $\frac{C^*}{C^A}=\infty>\frac{s+2}{s+1}$. If we assign J_2 to M_1, the third job J_3 of size $\frac{s}{s+1}$ comes. If J_3 is assigned to M_2, we have $C^A=\frac{1+s-s^2}{s^2+s}$, and $C^*=\frac{1}{s}$ by assigning J_1 to M_2 and J_2, J_3 to M_1. Hence $\frac{C^*}{C^A}=\frac{s+1}{1+s-s^2}>\frac{s+2}{s+1}$. Otherwise, if we assign J_3 to M_1, then the last job J_4 of size 1 comes. We obtain $C^A=\frac{1}{s}$, and $C^*=\frac{2s+1}{s^2+s}$ by assigning J_1, J_2 to M_1 and J_3, J_4 to M_2. Hence $\frac{C^*}{C^A}=\frac{2s+1}{s+1}\geqslant\frac{s+2}{s+1}$.

Lemma 2. The competitive ratio of any semi-online algorithm A for problem $Q2|\max|C_{\min}$ is at least $\min\left\{s, \frac{s+1}{s}\right\}$ when $\sqrt{2}< s< 1+\sqrt{2}$.

Proof. The first job J_1 has size $\frac{1}{s}$. We consider two cases.

Case 1. J_1 is assigned to M_1. The sequence continues with the largest job J_2 of size 1. If J_2 is also assigned to M_1, then no job comes. We get $C^A=0$, and $C^*=\frac{1}{s}$. It follows that $\frac{C^*}{C^A}=\infty>\min\left\{s, \frac{s+1}{s}\right\}$. If we assign J_2 to M_2, then the last job J_3 of size 1 comes. We get $C^A=\frac{1}{s}$, and $C^*=\min\left\{\frac{s+1}{s^2}, 1\right\}$ by assigning J_2 to M_1 and J_1, J_3 to M_2, then we have $\frac{C^*}{C^A}=\min\left\{s, \frac{s+1}{s}\right\}$.

Case 2. J_1 is assigned to M_2. The sequence continues with the largest job J_2 of size 1. It is obvious that J_2 must be assigned to M_1. So we get $C^A=\frac{1}{s^2}$, and $C^*=\frac{1}{s}$, then $\frac{C^*}{C^A}=s\geqslant\min\left\{s, \frac{s+1}{s}\right\}$.

Let $\alpha=\frac{1-s^2+\sqrt{s^4-s^2+2s+1}}{s}$ be the bigger root of equation $\frac{1+x}{sx}=\frac{2}{1+\frac{2}{s}-x}$ regarding x. Note that $0<\alpha\leqslant\frac{1}{s}$ when $s\geqslant1+\sqrt{2}$.

Lemma 3. The competitive ratio of any semi-online algorithm A for problem $Q2|\max|C_{\min}$ is at least $\frac{1+\alpha}{s\alpha}=\frac{s^2+s+1+\sqrt{s^4-s^2+2s+1}}{s^2+2s}$ when $s\geqslant 1+\sqrt{2}$.

Proof. The first job J_1 has size α. We consider two cases.

Case 1. J_1 is assigned to M_1. The sequence continues with the largest job J_2 of size 1. Obviously J_2 can not be assigned to M_1, so we assign J_2 to M_2, and J_3 of size 1 comes. If we also assign it to M_2, we can get $C^A=\alpha$ and $C^*=\frac{1+\alpha}{s}$ by assigning J_2 to M_1 and J_1, J_3 to M_2, then $\frac{C^*}{C^A}=\frac{1+\alpha}{s\alpha}$. If we assign J_3 to M_1, then the last job of size $\frac{2}{s}-\alpha$ comes. We have $C^A\leqslant\frac{1+\frac{2}{s}-\alpha}{s}$, and $C^*=\frac{2}{s}$ by assigning J_1, J_4 to M_1 and J_2, J_3 to M_2. Then it follows that $\frac{C^*}{C^A}=\frac{2}{1+\frac{2}{s}-\alpha}=\frac{1+\alpha}{s\alpha}$.

Case 2. J_1 is assigned to M_2. Then the last and the largest job comes. Obviously we must assign it to M_1, and get $C^A=\frac{\alpha}{s}$, $C^*=\alpha$. Hence $\frac{C^*}{C^A}=s>\frac{1+\alpha}{s\alpha}$ when $s\geqslant1+\sqrt{2}$.

2 Algorithms

In this section we will present two algorithms for $Q2|\max|C_{\min}$. Fast First List Scheduling (FFLS for short) and Slow First List Scheduling (SFLS for short) are designed for smaller and larger s, respectively. Both algorithms consist of two phases. In the second phase, they use LS rule to assign jobs, where LS rule always assigns jobs to the machine which can start to process the job earlier[9,6].

Denote by $J_{\max}$ the first job of size $p_{\max}=1$. Define the load of a machine as the total size of jobs assigned to it. Let $L(M_i)$ be the load of M_i after all the jobs are scheduled by a given algorithm A, $i=1, 2$. Therefore, $C^A=\min\left\{L(M_1), \frac{L(M_2)}{s}\right\}$. Note that $\frac{L(M_1)+L(M_2)}{s+1}$ is an upper bound on C^*. Hence, if $C^A=L(M_1)$, then

$$\frac{C^*}{C^A}\leqslant\frac{\frac{L(M_1)+L(M_2)}{s+1}}{L(M_1)}=\frac{1}{s+1}\left(1+\frac{L(M_2)}{L(M_1)}\right)$$

Otherwise, $C^A=\frac{L(M_2)}{s}$, then

$$\frac{C^*}{C^A}\leqslant\frac{\frac{L(M_1)+L(M_2)}{s+1}}{\frac{L(M_2)}{s}}=\frac{s}{s+1}\left(1+\frac{L(M_1)}{L(M_2)}\right)$$

The following lemma describes an important property of LS rule.

Lemma 4. (1) If $L(M_1)\geqslant\frac{L(M_2)}{s}$ and the last job on M_1 is assigned by LS rule, then $L(M_1) \leqslant \frac{L(M_2)}{s}+1$.

(2) If $\frac{L(M_2)}{s} \geqslant L(M_1)$ and the last job on M_2 is assigned by LS rule, then $L(M_2) \leqslant sL(M_1)+1$.

Proof. (1) Suppose the last job on M_1 is J_a of size p_a. Denote by $L^a(M_i)$ the load of M_i just before J_a is assigned, $i=1,2$. By LS rule, we have $L(M_1) - p_a = L^a(M_1) \leqslant \frac{L^a(M_2)}{s} \leqslant \frac{L(M_2)}{s}$, which implies that $L(M_1) \leqslant \frac{L(M_2)}{s} + p_a \leqslant \frac{L(M_2)}{s}+1$.

(2) Similar to (1), suppose the last job on M_2 is J_b of size p_b. Denote by $L^b(M_i)$ the load of M_i just before J_b is assigned, $i=1,2$. By LS rule, we have $\frac{L(M_2)-p_b}{s} = \frac{L^b(M_2)}{s} \leqslant L^b(M_1) \leqslant L(M_1)$, that is $L(M_2) \leqslant sL(M_1)+p_b \leqslant sL(M_1)+1$.

Let

$$\gamma_1 = \max\left\{s, \frac{s+2}{s+1}\right\} = \begin{cases} \frac{s+2}{s+1} & 1 \leqslant s \leqslant \sqrt{2} \\ s & \sqrt{2} < s \leqslant \frac{1+\sqrt{5}}{2} \end{cases}$$

Algorithm. FFLS.

Phase 1. Always assign current job J to M_2, unless one of the following two cases happens.

(1.1) J is $J_{\max}$. Then assign J to M_1, go to Phase 2.

(1.2) If J is assigned to M_2, the new load of M_2 will be greater than $\frac{s}{(s+1)(\gamma_1-1)}$, and J is not $J_{\max}$. Then assign J to M_2, go to Phase 2.

Phase 2. Assign all the remaining jobs by LS rule.

Lemma 5. $L(M_2) \geqslant \frac{1}{(s+1)(\gamma_1-1)}$.

Proof. If $J_{\max}$ is assigned to M_1, then $L(M_1) \geqslant 1 = \frac{1}{(s+1)\left(\frac{s+2}{s+1}-1\right)} \geqslant \frac{1}{(s+1)(\gamma_1-1)}$. Otherwise, $J_{\max}$ must be assigned to M_2 in Phase 2. Denote by $L^{\max}(M_i)$ the loads of M_i just before $J_{\max}$ is assigned, $i=1,2$. If $L^{\max}(M_2) < \frac{s}{(s+1)(\gamma_1-1)}$, $J_{\max}$ will be assigned to M_1 in Phase 1, which is a contradiction. Therefore, $L(M_1) \geqslant L^{\max}(M_1) \geqslant \frac{L^{\max}(M_2)}{s} \geqslant \frac{1}{(s+1)(\gamma_1-1)}$.

Theorem 1. The competitive ratio of the algorithm FFLS for $Q2|\max|C_{\min}$ when $1 < s \leqslant \frac{1+\sqrt{5}}{2}$ is at most γ_1.

Proof. We distinguish two cases according to the value of $L(M_2)$.

Case 1. $L(M_2) < \frac{s}{(s+1)(\gamma_1-1)}$.

In this case, $J_{\max}$ is assigned to M_1 in Phase 1, and it is the only job assigned to M_1 due to $\frac{L(M_2)}{s} < \frac{1}{(s+1)(\gamma_1-1)} \leqslant 1$. Therefore $L(M_1)=1$ and $C^{\mathrm{FFLS}} = \frac{L(M_2)}{s}$. If $L(M_2) \leqslant \frac{1}{s}$, then $\frac{C^*}{C^{\mathrm{FFLS}}} = \frac{L(M_2)}{\frac{L(M_2)}{s}} = s \leqslant \gamma_1$. If

$$\frac{1}{s} < L(M_2) < \frac{s}{(s+1)(\gamma_1-1)}$$

then

$$\frac{C^*}{C^{\mathrm{FFLS}}} \leqslant \frac{s}{s+1}\left(1+\frac{L(M_1)}{L(M_2)}\right) \leqslant \frac{s}{s+1}\left(1+\frac{1}{\frac{1}{s}}\right) = s \leqslant \gamma_1$$

Case 2. $L(M_2) \geqslant \frac{s}{(s+1)(\gamma_1-1)}$.

Subcase 2.1. $C^{\mathrm{FFLS}} = L(M_1)$.

If there is no job assigned to M_2 in Phase 2, then $L(M_2) < p_{\max} + \frac{s}{(s+1)(\gamma_1-1)} = 1 + \frac{s}{(s+1)(\gamma_1-1)}$, and $J_{\max}$ is assigned to M_1. Hence $L(M_1) \geqslant 1$, and

$$\frac{C^*}{C^{\mathrm{FFLS}}} \leqslant \frac{1}{s+1}\left(1+\frac{L(M_2)}{L(M_1)}\right) \leqslant \frac{1}{s+1}\left(1+1+\frac{s}{(s+1)(\gamma_1-1)}\right)$$

$$\leqslant \frac{1}{s+1}\left(2+\frac{s}{(s+1)\left(\frac{s+2}{s+1}-1\right)}\right)$$
$$=\frac{s+2}{s+1}\leqslant \gamma_1$$

If there are some jobs assigned to M_2 in Phase 2, by Lemmas 4(2) and 5, we have $L(M_2)\leqslant sL(M_1)+1\leqslant((s+1)\gamma_1-1)L(M_1)$. Hence

$$\frac{C^*}{C^{\mathrm{FFLS}}}\leqslant \frac{1}{s+1}\left(1+\frac{L(M_2)}{L(M_1)}\right)$$
$$\leqslant \frac{1}{s+1}(1+(s+1)\gamma_1-1)=\gamma_1$$

Subcase 2.2. $C^{\mathrm{FFLS}}=\frac{L(M_2)}{s}$.

If there are some jobs assigned to M_1 in Phase 2, we have $L(M_1)\leqslant\frac{L(M_2)}{s}+1$ by Lemma 4(1). If there is no job assigned to M_1 in Phase 2, $J_{\max}$ must be assigned to M_1 in Phase 1. In fact, by the description of FFLS, only $J_{\max}$ can be assigned to M_1 in Phase 1, and $J_{\max}$ can not be assigned to M_2 in Phase 1. If $J_{\max}$ is assigned to M_2 in Phase 2, it is assigned by LS rule, which contradicts to the fact that no job is assigned to M_1 when $J_{\max}$ comes. Therefore, we also have $L(M_1)=1\leqslant\frac{L(M_2)}{s}+1$. Hence

$$\frac{C^*}{C^{\mathrm{FFLS}}}\leqslant \frac{s}{s+1}\left(1+\frac{L(M_1)}{L(M_2)}\right)$$
$$\leqslant \frac{s}{s+1}\left(1+\left(\frac{L(M_2)}{s}+1\right)\frac{1}{L(M_2)}\right)$$
$$\leqslant \frac{s}{s+1}\left(1+\frac{1}{s}+\frac{(s+1)(\gamma_1-1)}{s}\right)=\gamma_1$$

The proof is thus finished.

Let

$$\gamma_2=\max\left\{\frac{s+1}{s},\frac{1+s+\sqrt{5s^2+6s+1}}{2(s+1)},\frac{1+s+s^2+\sqrt{s^4-s^2+2s+1}}{s(s+2)}\right\}$$
$$=\begin{cases}\frac{s+1}{s} & \frac{1+\sqrt{5}}{2}< s\leqslant s_1\\ \frac{1+s+\sqrt{5s^2+6s+1}}{2(s+1)} & s_1< s\leqslant s_2\\ \frac{1+s+s^2+\sqrt{s^4-s^2+2s+1}}{s(s+2)} & s> s_2\end{cases}$$

where $\frac{s+1+\sqrt{5s^2+6s+1}}{2(s+1)}$ is the biggest root of equation $\frac{s(s+1)x^2-sx-s^2}{(s+1)^2(x^2-x)-s}=x$ regarding x, and $\frac{s^2+s+1+\sqrt{s^4-s^2+2s+1}}{s(s+2)}$ is the bigger root of equation $\frac{1}{sx-1}=\frac{(s+2)x-2x}{sx}$ regarding x. We call J is a big job if J is not $J_{\max}$ and the size of J lies in the interval $\left[\frac{s+1}{s}\gamma_2-1-\frac{1}{(s+1)(\gamma_2-1)},1\right]$.

Algorithm. SFLS.

Phase 1.

(i) Always assign the current job J to M_1, unless the new load of M_1 will be greater than $\frac{1}{(s+1)(\gamma_2-1)}$ by assigning J to M_1.

(1.1) If by assigning J to M_1, the new load of M_1 would be in the interval

$$\left[\frac{1}{(s+1)(\gamma_2-1)},\frac{s+1}{s}\gamma_2-1\right]$$

then assign J to M_1, go to Phase 2.

(1.2) If by assigning J to M_1, the new load of M_1 would be greater than $\frac{s+1}{s}\gamma_2-1$, and J is $J_{\max}$, then assign J to M_2, return to Step 1 of Phase 1.

(1.3) If by assigning J to M_1, the new load of M_1 would be greater than $\frac{s+1}{s}\gamma_2-1$, and J is not $J_{\max}$, then go to Step 2.

(ii) If the current load of M_1 is less than $\frac{1}{s\gamma_2-1}$, then assign J to M_1, go to Phase 2. Otherwise, go to Step 3.

(iii) If there is already a big job on M_2, then assign J to M_1, go to Phase 2. Otherwise, assign J to M_2, return to Step 1 of Phase 1.

Phase 2. Assign all the remaining jobs by LS rule.

Note that

$$\frac{1}{s\gamma_2-1}<\frac{1}{(s+1)(\gamma_2-1)}<\frac{s+1}{s}\gamma_2-1$$

when $s>\frac{1+\sqrt{5}}{2}$, so the algorithm SFLS is well defined. Moreover, as

$$\frac{s+2}{s+1} < \frac{s+1}{s} \leqslant \gamma_2 < \frac{2s+1}{s+1}$$

$$\frac{1}{s} < \frac{1}{(s+1)(\gamma_2-1)} < 1$$

Theorem 2. The competitive ratio of the algorithm SFLS for $Q2|\max|C_{\min}$ when $s > \frac{1+\sqrt{5}}{2}$ is at most γ_2.

Proof. We distinguish two cases according to the value of $L(M_1)$.

Case 1. $L(M_1) < \frac{1}{(s+1)(\gamma_2-1)}$.

If $L(M_1) \leqslant \frac{1}{s}$, there is only one job $J_{\max}$ assigned to M_2, SFLS yields an optimal solution.

If $\frac{1}{s} < L(M_1) < \frac{1}{(s+1)(\gamma_2-1)}$, consider the jobs assigned to M_2. If there is only one job, $J_{\max}$, assigned to M_2, then

$$C^{\mathrm{SFLS}} = \frac{L(M_2)}{s} = \frac{1}{s} < L(M_1)$$

Hence

$$\frac{C^*}{C^{\mathrm{SFLS}}} \leqslant \frac{s}{s+1}\left(1+\frac{L(M_1)}{L(M_2)}\right) < \frac{s}{s+1}\left(1+\frac{1}{(s+1)(\gamma_2-1)}\right) \leqslant \gamma_2$$

The last inequality is due to

$$\gamma_2 \geqslant \frac{1+s+\sqrt{5s^2+6s+1}}{2(s+1)} \geqslant \frac{2s+1+\sqrt{4s+1}}{2(s+1)} \qquad (1)$$

Otherwise, we conclude that there must be two jobs, $J_{\max}$ and a big job, denoted by J' with size p', assigned to M_2. In fact, by the description of SFLS, the algorithm will not enter Phase 2 unless the current load of M_1 is greater than $\frac{1}{(s+1)(\gamma_2-1)}$. Moreover, M_2 processes at most two jobs in Phase 1. Hence

$$\begin{aligned} L(M_2) &= p'+1 \\ &\geqslant \left(\left(\frac{s+1}{s}\gamma_2-1\right)-\frac{1}{(s+1)(\gamma_2-1)}\right)+1 \\ &= \frac{s+1}{s}\gamma_2-\frac{1}{(s+1)(\gamma_2-1)} \qquad (2) \end{aligned}$$

If $C^{\mathrm{SFLS}} = \frac{L(M_2)}{s}$, by (1) and (2),

$$\begin{aligned} \frac{C^*}{C^{\mathrm{SFLS}}} &\leqslant \frac{s}{s+1}\left(1+\frac{L(M_1)}{L(M_2)}\right) \\ &\leqslant \frac{s}{s+1}\left(1+\frac{\frac{1}{(s+1)(\gamma_2-1)}}{\frac{s+1}{s}\gamma_2-\frac{1}{(s+1)(\gamma_2-1)}}\right) \\ &\leqslant \gamma_2 \end{aligned}$$

If $C^{\mathrm{SFLS}} = L(M_1)$, consider the assignment of $J_{\max}$ and J' in the optimal schedule. If $J_{\max}$ and J' are assigned to the same machine, then we can get $C^* \leqslant L(M_1) = C^{\mathrm{SFLS}}$, SFLS yields an optimal schedule. If $J_{\max}$ and J' are assigned to the different machines, we have $C^* \leqslant \frac{L(M_1)+p'}{s}$. Hence,

$$\frac{C^*}{C^{\mathrm{SFLS}}} \leqslant \frac{1}{s}\left(1+\frac{p'}{L(M_1)}\right) \leqslant \frac{1}{s}\left(1+\frac{1}{L(M_1)}\right) \leqslant \frac{s+1}{s} \leqslant \gamma_2$$

Case 2. $L(M_1) \geqslant \frac{1}{(s+1)(\gamma_2-1)}$.

In this case, algorithm SFLS must stop at Phase 2. We distinguish the three subcases based on the step by which Phase 1 enters Phase 2.

Subcase 2.1. Algorithm SFLS enters Phase 2 by Step (1.1). The load of M_1 at the beginning of Phase 2 is grater than $\frac{1}{(s+1)(\gamma_2-1)}$.

If $C^{\mathrm{SFLS}} = L(M_1)$ and there are jobs assigned to M_2 in Phase 2, from Lemma 4(2) we can see

$$\begin{aligned} L(M_2) &\leqslant sL(M_1)+1 \\ &\leqslant sL(M_1)+(s+1)(\gamma_2-1)L(M_1) \\ &= ((s+1)\gamma_2-1)L(M_1) \end{aligned}$$

Hence,

$$\frac{C^*}{C^{\mathrm{SFLS}}} \leqslant \frac{1}{s+1}\left(1+\frac{L(M_2)}{L(M_1)}\right) \leqslant \frac{1}{s+1}(1+((s+1)\gamma_2-1)) = \gamma_2$$

If $C^{\mathrm{SFLS}} = L(M_1)$ and there is no job assigned to M_2 in Phase 2, $J_{\max}$ and at most one big job are assigned to M_2 in Phase 1. Hence, $L(M_2) \leqslant 2$ and

$$\frac{C^*}{C^{\mathrm{SFLS}}} \leqslant \frac{1}{s+1}\left(1+\frac{L(M_2)}{L(M_1)}\right)$$

$$\leqslant \frac{1}{s+1}\left(1+\frac{2}{\frac{1}{(s+1)(\gamma_2-1)}}\right) \leqslant \gamma_2$$

The last inequality is due to $\gamma_2 \leqslant \frac{2s+1}{s+1}$.

If $C^{\mathrm{SFLS}} = \frac{L(M_2)}{s}$ and there is no job assigned to M_1 in Phase 2, we have

$$\frac{1}{(s+1)(\gamma_2-1)} \leqslant L(M_1) \leqslant \frac{s+1}{s}\gamma_2 - 1$$

and $L(M_2) \geqslant 1$, since $J_{\max}$ must be assigned to M_2. Therefore,

$$\frac{C^*}{C^{\mathrm{SFLS}}} \leqslant \frac{s}{s+1}\left(1+\frac{L(M_1)}{L(M_2)}\right) \leqslant \frac{s}{s+1}\left(1+\frac{s+1}{s}\gamma_2 - 1\right) = \gamma_2$$

If $C^{\mathrm{SFLS}} = \frac{L(M_2)}{s}$ and there are jobs assigned to M_1 in Phase 2, then $\frac{L(M_2)}{s} \geqslant \frac{1}{(s+1)(\gamma_2-1)}$ by LS rule. By Lemma 4(1), we have $L(M_1) \leqslant \frac{L(M_2)}{s} + 1$. Hence,

$$L(M_1) \leqslant \frac{1+(s+1)(\gamma_2-1)}{s} L(M_2)$$

and

$$\frac{C^*}{C^{\mathrm{SFLS}}} \leqslant \frac{s}{s+1}\left(1+\frac{L(M_1)}{L(M_2)}\right) \leqslant \frac{s}{s+1}\left(1+\frac{1+(s+1)(\gamma_2-1)}{s}\right) = \gamma_2$$

Subcase 2.2. Algorithm SFLS enters Phase 2 by Step 2 of Phase 1. The load of M_1 at the beginning of Phase 2 is grater than $\frac{s+1}{s}\gamma_2 - 1$.

If $C^{\mathrm{SFLS}} = L(M_1)$, by Lemma 4(1) we have

$$L(M_2) \leqslant sL(M_1) + 1 \leqslant \left(s + \frac{s}{(s+1)\gamma_2 - s}\right) L(M_1)$$

Hence, by (1),

$$\frac{C^*}{C^{\mathrm{SFLS}}} \leqslant \frac{1}{s+1}\left(1+\frac{L(M_2)}{L(M_1)}\right) \leqslant \frac{1}{s+1}\left(1+s+\frac{s}{(s+1)\gamma_2 - s}\right) \leqslant \gamma_2$$

If $C^{\mathrm{SFLS}} = \frac{L(M_2)}{s}$ and there are jobs assigned to M_1 in Phase 2, then

$$\frac{L(M_2)}{s} \geqslant \frac{s+1}{s}\gamma_2 - 1$$

By Lemma 4(1), we have

$$L(M_1) \leqslant \frac{L(M_2)}{s} + 1 \leqslant \left(\frac{1}{s} + \frac{1}{(s+1)\gamma_2 - s}\right) L(M_2)$$

Combining with (1),

$$\frac{C^*}{C^{\mathrm{SFLS}}} \leqslant \frac{s}{s+1}\left(1+\frac{L(M_1)}{L(M_2)}\right) \leqslant \frac{s}{s+1}\left(1+\frac{1}{s}+\frac{1}{(s+1)\gamma_2 - s}\right) \leqslant \gamma_2$$

If $C^{\mathrm{SFLS}} = \frac{L(M_2)}{s}$ and there is no job assigned to M_1 in Phase 2, then the last job assigned to M_1 is a big job J'' of size p''. Denote by $L''(M_1)$ the load of M_1 just before J'' is assigned. Then $L''(M_1) \leqslant \frac{1}{s\gamma_2 - 1}$ and $L(M_1) = L''(M_1) + p''$. Note that $J_{\max}$ is assigned to M_2. Let the total size of jobs assigned to M_2 other than $J_{\max}$ be $L''(M_2)$, i.e., $L(M_2) = L''(M_2) + 1$.

If $L(M_1) \leqslant \left(\frac{s+1}{s}\gamma_2 - 1\right) L(M_2)$, then

$$\frac{C^*}{C^{\mathrm{SFLS}}} \leqslant \frac{s}{s+1}\left(1+\frac{L(M_1)}{L(M_2)}\right) \leqslant \frac{s}{s+1}\left(1+\frac{s+1}{s}\gamma_2 - 1\right) = \gamma_2$$

Otherwise,

$$L(M_1) > \left(\frac{s+1}{s}\gamma_2 - 1\right) L(M_2)$$

Consider the assignment of $J_{\max}$ and J'' in the optimal schedule. If $J_{\max}$ and J'' are assigned to the different machines,

$$C^* \leqslant \frac{L''(M_1) + L''(M_2) + 1}{s}$$

Since

$$L''(M_1) \leqslant \frac{1}{s\gamma_2 - 1} \leqslant \frac{1}{s} \leqslant \frac{1}{s}(1 + L''(M_2))$$

$$\frac{C^*}{C^{\mathrm{SFLS}}} \leqslant \frac{L''(M_1) + L''(M_2) + 1}{L''(M_2) + 1} \leqslant 1 + \frac{1}{s} \leqslant \gamma_2$$

If $J_{\max}$ and J'' are assigned to the same machine in the optimal schedule, we have

$$C^* \leqslant L''(M_1) + L''(M_2)$$

Since

$$\left(\frac{s+1}{s}\gamma_2 - 1\right)(L''(M_2) + 1) = \left(\frac{s+1}{s}\gamma_2 - 1\right) L(M_2) < L(M_1)$$

$$= L''(M_1) + p'' \leqslant L''(M_1) + 1$$

and

$$L''(M_1) < \frac{1}{s\gamma_2 - 1} \leqslant \frac{(s+2)\gamma_2 - 2s}{s\gamma_2}$$

due to $\gamma_2 \geqslant \frac{s^2+s+1+\sqrt{s^4-s^2+2s+1}}{s(s+2)}$, we have

$$\begin{aligned} &sL''(M_1) + (s-\gamma_2)L''(M_2) \\ &\leqslant sL''(M_1) + (s-\gamma_2)\left(\frac{s(L''(M_1)+1)}{(s+1)\gamma_2 - s} - 1\right) \\ &= s^2\gamma_2 \frac{sL''(M_1)+1}{(s+1)\gamma_2 - s} - 2s + \gamma_2 \\ &\leqslant s^2\gamma_2 \frac{\frac{(s+2)\gamma_2 - 2s}{s\gamma_2} + 1}{(s+1)\gamma_2 - s} - 2s + \gamma_2 \\ &= \frac{s(s+2)\gamma_2 - 2s^2 + s^2\gamma_2}{(s+1)\gamma_2 - s} - 2s + \gamma_2 = \gamma_2 \end{aligned}$$

Hence, $\frac{C^*}{C^{\mathrm{SFLS}}} \leqslant \frac{s(L''(M_1)+L''(M_2))}{1+L''(M_2)} \leqslant \gamma_2$.

Subcase 2.3. Algorithm SFLS enters Phase 2 by Step 3 of Phase 1.

If $C^{\mathrm{SFLS}} = L(M_1)$, or $C^{\mathrm{SFLS}} = \frac{L(M_2)}{s}$ and there are jobs assigned to M_1 in Phase 2, the proof is the same as Subcase 2.2.

If $C^{\mathrm{SFLS}} = \frac{L(M_2)}{s}$, and there is no job assigned to M_1 in Phase 2, then $L(M_1) < \frac{1}{(s+1)(\gamma_2 - 1)} + 1$. By the description of SFLS, we know that a big job J' of size p' is assigned to M_2 in Phase 1. And $J_{\max}$, no matter whether it is assigned in Phase 1 or Phase 2, is also assigned to M_2. Therefore,

$$\begin{aligned} L(M_2) &\geqslant p' + 1 \\ &\geqslant \frac{s+1}{s}\gamma_2 - 1 - \frac{1}{(s+1)(\gamma_2-1)} + 1 \\ &= \frac{s+1}{s}\gamma_2 - \frac{1}{(s+1)(\gamma_2-1)} \end{aligned}$$

and we obtain

$$\begin{aligned} \frac{C^*}{C^{\mathrm{SFLS}}} &= \frac{s}{s+1}\left(1 + \frac{L(M_1)}{L(M_2)}\right) \\ &\leqslant \frac{s}{s+1}\left(1 + \frac{1 + \frac{1}{(s+1)(\gamma_2-1)}}{\frac{s+1}{s}\gamma_2 - \frac{1}{(s+1)(\gamma_2-1)}}\right) \\ &\leqslant \gamma_2 \end{aligned}$$

where the last inequality is equivalent to

$$((s+1)\gamma_2 - s)((s+1)\gamma_2^2 - (s+1)\gamma_2 - s) \geqslant 0$$

which is valid by the definition of γ_2. The proof is thus finished.

By Theorems 4 and 5, we know that FFLS is an optimal algorithm for $1 \leqslant s \leqslant \frac{1+\sqrt{5}}{2}$, and SFLS is an optimal algorithm for $s \in [1.618, 2.148] \cup [3.836, \infty)$. For the interval $(2.148, 3.836)$ in which SFLS is not optimal, the competitive ratio of SFLS is monotone increasing. On the other hand, it can be verified directly that $\frac{s+1}{s}$ is monotone decreasing when

$$s \in [2.148, 2.414]$$

and $\frac{s^2+s+1+\sqrt{s^4-s^2+2s+1}}{s^2+2s}$ is monotone increasing when $s \in [2.414, 3.836]$. Hence, the largest gap between the competitive ratio and the lower bound for any s is $\frac{\sqrt{1+2\sqrt{2}}+1-2\sqrt{2}}{2} \approx 0.064$, which achieves at $1+\sqrt{2} \approx 2.414$. Moreover, the overall competitive ratio 2, which achieves when s tends to ∞, also matches the overall lower bound.

Acknowledgement The authors would like to acknowledge an anonymous referee for his careful reading of the paper and helpful comments.

References

1 Deuermeyer B, Friesen D, Langston M. Scheduling to maximize the minimum processor finish time in a multiprocessor system. SIAM Journal on Algebraic and Discrete Methods, 1982, 3: 190—196

2 Epstein L. Tight bounds for on-line bandwidth allocation on two links. Discrete Applied Mathematics, 2005, 148: 181—188

3 Kellerer H, Kotov V, Speranza M R, et al. Semi on-line algorithms for the partition problem. Operations Research Letters, 1997, 21: 235—242

4 He Y, Zhang GC. Semi on-line scheduling on two identical machines. Computing, 1999, 62: 179—187

5 Azar Y, Regev O. On-line bin-stretching. Theoretical Computer Science, 2001, 168: 17—41

6 Cheng TCE, Kellerer H, Kotov V. Semi-on-line multiprocessor scheduling with given total processing time. Theoretical Computer Science, 2005, 337: 134—146

7 Tan ZY, Cao SJ. Semi-online machine covering on two uniform machines with known total size. Computing, 2006, 78: 369—378

8 Luo RZ, Sun SJ, Huang WP. Semi on-line scheduling problem for maximizing the minimum machine completion time on two uniform machines. Journal of Systems Science & Complexcity, 2006, 19: 101—107

9 Graham RL. Bounds for certain multiprocessor anomalies. Bell Systems Technical Journal, 1966, 45: 1563—1581