On-line Scheduling of Two Parallel Machines with a Single Server

Lele Zhang, Andrew Wirth*

Department of Mechanical and Manufacturing Engineering, The University of Melbourne, VIC 3010, Australia

Abstract

In this paper, we consider the on-line scheduling of two parallel identical machines sharing a single server with the objective of minimizing the latest completion time of all jobs. Each job has to be setup by the server before being processed on one of the machines. Three special cases: equal length jobs, equal processing times and regular equal setup times are considered and the asymptotic competitive ratios of list scheduling are determined. Also, a lower bound for the equal length job case is given, and two heuristics with tight asymptotic competitive ratios for the other two cases are proposed.

Key words: On-line scheduling; Parallel machines; Server; List scheduling

1 Introduction

In many manufacturing situations, some pre-operational work (e.g. loading) needs to be executed before the processing of jobs. We consider the problem

* Corresponding author. Tel.: +61-3-8344 4852; Fax: +61-3-9347 8784

Email address: wirtha@unimelb.edu.au (Andrew Wirth).

Preprint submitted to Elsevier 17 August 2007
of scheduling parallel machines with setups. More specifically, this paper considers the on-line scheduling of two identical parallel machines with a single server, where \( n \) jobs must be processed non-preemptively on either of the machines after the completion of their setups on the server. The server can handle at most one job at a time. The objective is to minimize the latest completion time of all jobs. In the standard notation \([10]\), our problem could be denoted by \( P_2, S_1|s_i|C_{\text{max}} \).

One application of the problem, which is mentioned in \([8]\), is a robot system where a robot is shared by two semiautomatic machines. There have been several papers discussing the deterministic (off-line) version of this problem. Koulamas \([8]\) reduced the problem to a smaller one with all consecutive jobs processed on machines alternately and proposed a beam search heuristic with favorable computational performance, based on the reduction. Kravchenko and Werner \([10]\) presented a pseudo-polynomial algorithm for the unit setup time problem and derived the problem’s computational complexity for an arbitrary number of machines. They also analyzed a list scheduling heuristic for this problem. In subsequent work, Hall et al. \([7]\) provided a comprehensive study of the computational complexity of the problem for various classical scheduling objectives, as well as some polynomial or pseudo-polynomial algorithms for them. The work of Abdekhodaei et al. \([2–4]\), which is closer to the issues discussed in this paper, resolved the computational complexity of the problem and provided some effective heuristics for some special and general cases.

However, all of the above articles have concentrated on the off-line version of the scheduling, where the scheduler must have full access to the information of the problem before the scheduling starts. In fact, in many real life situations, only partial of information is available in advance, while decision-making is required immediately. In this paper, we apply list scheduling (LS) \([6]\), the earliest and simplest on-line algorithm, to three special cases of the two parallel machines with a single server problem - the equal length problem (ELP), the equal processing time problem (EPP) and the regular equal setup time problem (RESP), and establish the asymptotic competitive ratio of LS for each case. Furthermore, we derive a lower bound of any on-line algorithm for ELP and also propose two on-line heuristics for EPP and RESP, which are shown to have better asymptotic performance than LS.

The remainder of the paper is organized as follows. Section 2 presents the
problem model and recalls some useful notation and propositions from previous work. Section 3 is dedicated to ELP, where an on-line lower bound and the asymptotic competitive ratio of LS are given. In Sections 4 and 5, we propose on-line heuristics for EPP and RESP, respectively, and analyze the asymptotic performance of LS and the new heuristics. Finally, in Section 6 we draw some conclusions and present some suggestions for future work.

2 Model

Notation:

First we recall some notation from [2,3]:

- $M_1$: machine 1 which processes the first job;
- $M_2$: machine 2;
- $n$: number of jobs;
- $s_i$: setup time of job $i$ in a sequence;
- $p_i$: processing time of job $i$ in a sequence;
- $a_i$: length of job $i$, $a_i = s_i + p_i$;
- $S_i$: setup time of the $i$th job in a schedule;
- $P_i$: processing time of the $i$th job in a schedule;
- $C_i$: completion time of the $i$th scheduled job;
- $I_i$: the $i$th machine idle time;
- $W_i$: server waiting time between the $(i + 1)$th and $(i + 2)$th scheduled jobs;

In this paper we shall assume that we have a list of $n$ jobs, available at time zero, such that $p_i \leq a_j \forall i, j$. That is, the job set is regular, [2], and so any off-line schedule not allowing unnecessary idle time will process the jobs alternately on the two machines. We also assume, for now, that we schedule the jobs on-line, one-by-one (on-line list), Pruhs et al. [9]. That is, we must
allocate machines and starting times to the jobs, one by one, without any knowledge of the remaining jobs in the list.

Two evaluation measures for an on-line algorithm $A$ are its competitive performance ratio $R_A = \inf \{r \geq 1 : A(I)/OPT(I) \leq r, \text{ for all } I \}$, where $OPT$ denotes the optimal off-line solution, and $I$ is a problem instance and asymptotic performance ratio $R_A^\infty = \inf \{r \geq 1 : \text{ for some } N > 0, A(I)/OPT(I) \leq r, \text{ for all } I \text{ with } OPT(I) \geq N \}$. The latter measure may avoid certain small case anomalies.

In some of the proofs presented below we make use of jobs with very small processing or setup times. That is, we may have $p_i = \epsilon$ or $s_i = \epsilon$ where $\epsilon$ is a small positive number and then let $\epsilon \to 0$. By a slight abuse of language we shall abbreviate this process by saying that we have a set of jobs with zero processing or setup times.

We now recall the following propositions from [2]:

**Proposition 1** Under the assumption of alternating processing and regular jobs, $W_i = T_{i+2} - T_{i+1} - S_{i+1} = (P_i - S_{i+1} - W_{i-1})^+$ for $0 \leq i \leq n$, $I_i = (T_{i+1} + S_{i+1} - T_i - S_i + P_i)^+$ = $(P_i - S_{i+1} - W_{i-1})^+$ for $0 \leq i \leq n + 1$, $T_{i+2} = T_i + I_i + S_i + P_i$ for $0 \leq i \leq n$ and $W_i \wedge I_i = 0$ for $0 \leq i \leq n$.

**Proposition 2** Under the assumption of alternating processing and regular jobs, $C_{\text{max}} = \sum_{i=1}^{n} s_i + \sum_{i=1}^{n} W_i = 1/2(\sum_{i=1}^{n} a_i + \sum_{i=0}^{n+1} I_i)$.

**Proposition 3** If $p_j \leq s_k$ for $\forall j, k$, then $W_i = 0$ for $1 \leq n - 1$ and $W_n = P_n$.

3 **Equal length problem** ($P_2, S_1|a_i = a|C_{\text{max}}$)

3.1 **A lower bound of competitive ratio**

The off-line version of this problem can be solved optimally with an $O(n \log n)$ time algorithm, [2]. The equal length problem has a close link with classic bin packing. Here we modify slightly an on-line bin packing argument Liang [11] to obtain a lower bound for the asymptotic competitive ratio.

**Proposition 4** For the problem $P_2, S_1|a_i = a|C_{\text{max}}$, no on-line algorithm can
have an asymptotic ratio less than 4/3.

**Proof.** Suppose, without loss of generality, that $a = 1$. Consider a sequence $\sigma$ of $n$ jobs with zero setup time. Suppose an algorithm $A$ completes the sequence at time $c$. Then a sequence $\sigma'$ of $n$ jobs with zero processing time is presented. At best, $A$ will complete $2c - n$ of additional jobs by time $c$. So $A(\sigma + \sigma') \geq c + n - (2c - n) = 2n - c$. Assume that the asymptotic competitive ratio is $\rho$. We have that $c \leq \rho\text{OPT}(\sigma) = \rho n/2$ and $2n - c \leq \rho\text{OPT}(\sigma + \sigma') = \rho n$. Thus, $2n - \rho n/2 \leq 2n - c \leq \rho n$ and $\rho \geq 4/3$. The result follows. \(\square\)

### 3.2 List Scheduling

Graham [6] introduced the classic list scheduling on-line algorithm. Each job is scheduled to the first available machine at the earliest possible time.

For ELP, let $L$ represent a job with long setup ($s_i > a/2$) and $R$ represent a job with short setup ($s_i \leq a/2$).

**Proposition 5** A sequence in which the first job has equal setup and processing times and the following jobs are all short setup jobs, that is, $R_1 R_2 R_3 \cdots R_n$ in which $S_1 = a/2$ and $S_i \leq a/2$ for $2 \leq i \leq n$, has $T_{i+1} - T_i = a/2$ for $1 \leq i \leq n$.

**Proof.** Since $T_1 = 0$ and $T_2 = S_1 = a/2$, $T_2 - T_1 = a/2$. Assume $T_{k+1} - T_k = a/2$ for $2 \leq k \leq n - 1$.

$$T_{k+2} = \max(T_{k+1} + S_{k+1}, T_k + a) = \max(T_{k+1} + S_{k+1}, T_{k+1} + a/2) = T_{k+1} + a/2$$

since $S_{k+1} \leq a/2$. Hence the result follows. \(\square\)

**Lemma 6** If the sequence of Proposition 5 is within another sequence, that is,$$
\cdots L_i R_{i+1} R_{i+2} \cdots R_k L_{k+1} \cdots
$$
then $T_{j+1} = T_j + a/2$ for $i \leq j \leq k$ subject to $T_{i+1} = T_i + a/2$.

**Proof.** Since $T_{i+1} - T_i = a/2$ and $S_j \leq a/2$ for $i \leq j \leq k$, the proof is similar to Proposition 5’s. The result follows. \(\square\)
Proposition 7  $R_{LS}^\infty = 3/2$ for the equal length problem.

Proof. Consider an arbitrary sequence $\sigma$ of equal length jobs. Its asymptotic competitive ratio can be obtained by following steps.

Step: 1 We add a job $M$, whose setup and processing times are equal, that is $S_M = a/2$, and put it at the beginning of the sequence. Denote the new sequence by $\sigma'$. We will prove that the makespan of sequence $\sigma'$ does not decrease after it is sorted in non-increasing setup time order.

Step: 2 If the job following job $M$ is a long setup job, move it to the first position of the sequence. Otherwise, check each successive job until the first long setup job comes up and move it to the first position. Continue the procedure until all long setup jobs after job $M$ are moved.

Now let us look on the change of makespan for each move. Suppose there are $p-1$ long setup jobs in front of job $M$ and $q$ jobs which form a run of purely short setup jobs following job $M$. If $p = 1$, it means no long setup job has been moved. The sequence is represented by

$$L_1L_2\cdots L_{p-1}R_MR_{p+1}R_{p+2}\cdots R_{p+q}L_{p+q+1}\cdots$$

for $p \geq 1$ and $q \geq 0$. We show that moving job $p + q + 1$, whose setup time is greater than $a/2$, to the first position does not reduce the makespan. The move could be separated into two substeps as follows, Fig. 1.

**Insert Figure 1 here**

Substep 1: Remove job $p + q + 1$ from the sequence.

Since $S_i > a/2$ for $1 \leq i \leq p$, by Proposition 3,

$$T_p = \begin{cases} 
0 & \text{if } p = 1 \\
\sum_{i=1}^{p-1} S_i & \text{if } p \geq 2
\end{cases}$$

$$T_{p+1} = \sum_{i=1}^{p} S_i$$
\[ T_{p+1} - T_p = \sum_{i=1}^{p} S_i - \sum_{i=1}^{p-1} S_i = S_p = S_M = a/2 \]

By Lemma 6, \( T_{p+q+1} - T_{p+q} = a/2 \). Thus, before the remove the start times of jobs \( p + q + 2 \) and \( p + q + 3 \) are given by

\[
T_{p+q+2} = \max(T_{p+q+1} + S_{p+q+1}, T_{p+q} + a) = T_{p+q+1} + S_{p+q+1} \\
T_{p+q+3} = \max(T_{p+q+2} + S_{p+q+2}, T_{p+q+1} + a) \\
= \max(T_{p+q+1} + S_{p+q+1} + S_{p+q+2}, T_{p+q+1} + a)
\]

since \( S_{p+q+1} > a/2 \).

After the remove, the start times are given by

\[
T'_{p+q+2} = T_{p+q+1} \\
T'_{p+q+3} = \max(T_{p+q+1} + S_{p+q+2}, T_{p+q} + a) \\
= \max(T_{p+q+1} + S_{p+q+2}, T_{p+q+1} + a/2)
\]

**Substep 2:** Insert job \( p + q + 1 \) to the first position of the sequence.

Since the first two jobs after the insert are long setup jobs, the setup time increments of jobs 2 and 3 both equal to \( S''_1 \), which is equal to \( S_{p+q+1} \). That is to say the whole sequence is shifted right by \( S_{p+q+1} \) without changing the processes of other jobs. Note that, for the first switch, since the setup time of job \( M \) (a short setup job) is equal to \( 1/2 \), the result above is the same. Thus the start times of jobs \( p + q + 2 \) and \( p + q + 3 \) are given by

\[
T''_{p+q+2} = T'_{p+q+2} + S_{p+q+1} = T_{p+q+1} + S_{p+q+1} \\
T''_{p+q+3} = T'_{p+q+3} + S_{p+q+1} \\
= \max(T_{p+q+1} + S_{p+q+2} + S_{p+q+1}, T_{p+q+1} + a/2 + S_{p+q+1})
\]

Consider the changes of the start times. Since

\[
T''_{p+q+2} - T'_{p+q+2} = T_{p+q+1} + S_{p+q+1} - T_{p+q+1} - S_{p+q+1} = 0
\]

the start time of job \( p + q + 2 \) does not change. As for job \( p + q + 3 \),
$$T''_{p+q+3} - T_{p+q+3} = \begin{cases} 
T_{p+q+1} + S_{p+q+1} + S_{p+q+2} & \text{if } S_{p+q+2} > a/2 \\
-(T_{p+q+1} + S_{p+q+1} + S_{p+q+2}) & \\
T_{p+q+1} + S_{p+q+1} + a/2 & \text{if } S_{p+q+2} \leq a/2 \text{ and } S_{p+q+1} + S_{p+q+2} > a \\
-(T_{p+q+1} + S_{p+q+1} + S_{p+q+2}) & \\
T_{p+q+1} + S_{p+q+1} + a/2 & \text{if } S_{p+q+2} \leq a/2 \text{ and } S_{p+q+1} + S_{p+q+2} \leq a \\
-(T_{p+q+1} + a) & \\
0 & \text{if } S_{p+q+2} > a/2 \\
a/2 - S_{p+q+2} \geq 0 & \text{if } S_{p+q+2} \leq a/2 \text{ and } S_{p+q+1} + S_{p+q+2} > a \\
S_{p+q+1} - a/2 \geq 0 & \text{if } S_{p+q+2} \leq a/2 \text{ and } S_{p+q+1} + S_{p+q+2} \leq a 
\end{cases}$$

since $S_{p+q+1} > a/2$.

Hence the makespan of the sequence is not reduced after each move of long setup jobs. Obviously, the makespan does not decrease after all moves.

**Step: 3** Suppose there are $l$ long setup jobs and $r$ short setup jobs in the sequence, excluding job $M$ which is in the $l + 1$th position. Sort all long setup jobs before job $M$ in non-increasing setup time order. Since $S_i > a/2$ for $1 \leq i \leq l$, by Proposition 3, $T_{l+1} = \sum_{i=1}^{l} S_i$ and $T_{l+2} = \sum_{i=1}^{l+1} S_i$. The makespan does not change after the sort.

Then sort all short setup jobs after job $M$ in non-increasing setup time order. By Lemma 6, we have that the makespan does not change either. The sequence is finally sorted in non-increasing setup time order, since $S_1 \geq S_2 \cdots \geq S_l > S_M \geq S_{l+1} \cdots \geq S_{l+r+1}$, without changing the makespan.

Denote the sequence in non-increasing setup time order by $\sigma''$. Delete the added job $M$ from the sequence $\sigma''$. The final sequence denoted by $\sigma_s$ is presented by

$$L_1L_2\cdots L_lS_{l+1}R_{l+2}\cdots R_{l+r}$$
where $S_i \geq S_{i+1}$ for $1 \leq i \leq l + r - 1$. Obviously, $\sigma$ and $\sigma_s$ have the same job composition. Thus $OPT(\sigma) = OPT(\sigma_s)$. Besides, $C_{\max}(\sigma') \geq C_{\max}(\sigma) + a/2$ and $C_{\max}(\sigma_s) + a \geq C_{\max}(\sigma'')$ as well as $C_{\max}(\sigma'') \geq C_{\max}(\sigma')$, which is proved in step 2. It can be concluded that

$$C_{\max}(\sigma) \leq C_{\max}(\sigma_s) + a/2$$

**Step 4:** Find the maximum value of $C_{\max}(\sigma)/OPT(\sigma)$.

**Case (i):** $l \leq r$

First we consider the makespans of long setup and short setup jobs, respectively.

For long setup jobs, $a/2 < S_i \leq a$ for $1 \leq i \leq l$, by Proposition 3,

$$C_{\max}(L) = \sum_{i=1}^{l} S_i + P_i \leq la + a/2$$

For short setup jobs, $0 \leq S_i \leq a/2$ for $l + 1 \leq i \leq l + r$, by Lemma 6

$$C_{\max}(R) = T_{l+r+2} - T_{l+1} \leq \frac{r+1}{2}a$$

Then, as to the makespan of the whole sequence $\sigma_s$,

$$C_{\max}(\sigma_s) \leq C_{\max}(L) + C_{\max}(R) \leq la + \frac{a}{2} + \frac{r+1}{2}a$$

$$C_{\max}(\sigma) \leq C_{\max}(\sigma_s) + a/2 \leq la + \frac{3a}{2} + \frac{r}{2}a$$

$$OPT(\sigma) = OPT(\sigma_s) = \frac{1}{2} \left( \sum_{i=1}^{l+r} a_i + \sum_{i=0}^{l+r+1} I_i \right) \geq \frac{r + a}{2}$$

$$\frac{C_{\max}(\sigma)}{OPT(\sigma)} \leq \frac{(l + r/2 + 3/2)a}{(l + r)a/2} = \frac{2l/r + 1 + 3/r}{l/r + 1}$$

$$\rightarrow \frac{2l/r + 1}{l/r + 1} \quad \text{as} \quad r \rightarrow \infty$$

$$= 2 - \frac{1}{l/r + 1}$$
Now $l/r \leq 1$, so by simple calculus, the fraction is maximized when $l/r = 1$. Thus, $C_{\text{max}}(\sigma)/\text{OPT}(\sigma) \leq 3/2$. Hence we have the asymptotic competitive ratio equals to $3/2$ for $l \leq r$.

**Case (ii): $l \geq r$**

In order to simplify the following proof we consider the $l$ long setup jobs as two parts: the first $r + 1$ jobs and the following $l - r - 1$ jobs. Then we have

$$C_{\text{max}}(L) = \sum_{i=1}^{r+1} S_i + \sum_{i=r+2}^{l} S_i + P_i \leq (r+1)a + \sum_{i=r+2}^{l} S_i + \frac{a}{2}$$

$$C_{\text{max}}(\sigma) \leq C_{\text{max}}(L) + C_{\text{max}}(R) \leq (r+1)a + \frac{a}{2} + \sum_{i=r+2}^{l} S_i + \frac{r+1}{2}a$$

$$= \frac{3a(r+1)}{2} + \frac{a}{2} + \sum_{i=r+2}^{l} S_i$$

$$C_{\text{max}}(\sigma) \leq \frac{3a(r+1)}{2} + a + \sum_{i=r+2}^{l} S_i$$

Recall the Lemma 9 in [2].

$$C_{\text{max}} \geq s_j + (s_{j+1} + s_{j-1} - a)^+ + \cdots + (s_{2j-1} + s_1 - a)^+ + ja$$

where the sequence is $s_1s_2\cdots s_j s_{j+1}\cdots s_{2j}$ and subject to $s_i \leq s_{i+1}$ for $1 \leq i \leq 2j - 1$.

Without loss of generality, we assume $l + r$ is even, that is, $l + r = 2j$. Since $l \geq r$ and $l \geq j$, the sequence is given by

$$L_1L_2\cdots L_{2j-i-1}\cdots L_j L_{j+1}\cdots L_l R_{l+1}\cdots R_{l+r}$$

where $S_i \geq S_{i+1}$ for $1 \leq i \leq l + r - 1$ and $S_i \geq a/2$ for $1 \leq i \leq l$. So we have $S_l + S_{2j-l+2} \geq a$, $\cdots$, and $S_j + S_{j+2} \geq a$, as Fig. 2 shows.

**Insert Figure 2 here**

Thus

$$\text{OPT}(\sigma) = \text{OPT}(\sigma_s)$$
\[ \geq S_{j+1} + (S_{j+2} + S_j - a) + \cdots + (S_{2j-l+2} + S_l - a) + ja \]
\[ = \sum_{i=r+2}^{l} S_i - \frac{(l-r-2)a}{2} + \frac{(l+r)a}{2} = \sum_{i=r+2}^{l} S_i + (r+1)a \]

Then

\[ \frac{C_{\text{max}}(\sigma)}{OPT(\sigma)} \leq \frac{3a(r+1)/2 + a + \sum_{i=r+2}^{l} S_i}{(r+1)a + \sum_{i=r+2}^{l} S_i} \]

Denote \((r + 1)a\) and \(\sum_{i=r+2}^{l} S_i\) by \(\alpha\) and \(\beta\), respectively. Thus,

\[ \frac{C_{\text{max}}(\sigma)}{OPT(\sigma)} \leq \frac{3\alpha/2 + a + \beta}{\alpha + \beta} \]
\[ - \frac{3\alpha/2 + \beta}{\alpha + \beta} \quad \text{as } l \to \infty, \quad \frac{a}{\alpha+\beta} \to 0 \]
\[ = 1 + \frac{1/2}{1 + \beta/\alpha} \leq \frac{3}{2} \]

So \(R_{LS}^\infty \leq 3/2\).

Consider the following sequence: the first \(n\) jobs, with even \(n\), have \(P_i = \epsilon, s_i = 1 - \epsilon\) and the next \(n\) jobs have \(p_i = 1 - \epsilon, s_i = \epsilon\). The list scheduling as well as its corresponding optimal solution are shown in Fig. 3. Clearly, \(C_{\text{max}}(LS) \to 3n/2\) and \(OPT \to n/2\) as \(\epsilon \to 0\). The sequence achieves the upper bound of asymptotic competitive ratio for the equal length problem. \(\square\)

**Insert Figure 3 here**

4 EPP \((P_2, S_1|p_i = p|C_{\text{max}})\)

We note that the off-line version of this problem is NP-hard, Abdekhodaee et al. [3]. Now we introduce some new notation within this section: Job subsets: \(L = \{\text{job } i : s_i > p\}\) and \(R = \{\text{job } i : s_i \leq p\}\). Let an index (1 or 2) indicates the number of the machine which processes the jobs in the subset. Let lower cases represent the numbers of jobs in each subset, thus \(|L| = l\) and \(|R| = r\).
4.1 LS

Rearrange a list schedule $\sigma$ in following way: If $S_{2i-1} < S_{2i}$, swap jobs $2i-1$ and $2i$ for $1 \leq i \leq \lfloor n/2 \rfloor$.

Denote the new schedule after the rearrangement by $\sigma'$ and the setup time of the $i$th processed job in $\sigma'$ by $S'_i$. Clearly, $S'_{2i-1} \geq S'_{2i}$ for $1 \leq i \leq \lfloor n/2 \rfloor$.

Firstly, we consider the short setup problem.

Lemma 8 $T'_{2i+1} = C'_{2i+1} = T'_{2i-1} + S'_{2i-1} + p$, for $1 \leq i \leq \lfloor n/2 \rfloor$, for the short setup problem.

Proof. For $1 \leq i \leq \lfloor n/2 \rfloor$,

$$T'_{2i+1} = \max(T'_2 + S'_2, T'_{2i-1} + S'_{2i-1} + p)$$
$$= \max(T'_{2i-1} + S'_{2i-1} + S'_2, T'_{2i-2} + S'_{2i-2} + S'_2 + p, T'_{2i-1} + S'_{2i-1} + p)$$
$$= \max(T'_{2i-2} + S'_{2i-2} + S'_2 + p, T'_{2i-1} + S'_{2i-1} + p)$$
$$= T'_{2i-1} + S'_{2i-1} + p$$

since $p \geq S'_{2i-1} \geq S'_2$ and $T'_{2i-1} \geq T'_{2i-2} + S'_{2i-2}$. Hence the result follows. \(\square\)

Lemma 9 $C_{\text{max}}(\sigma) \leq C_{\text{max}}(\sigma') + p$

Proof. We start by showing, by induction, that $T_{2i-1} \leq T'_{2i-1}$ for $1 \leq i \leq \lfloor n/2 \rfloor + 1$. The start times of the first three jobs of $\sigma$ and $\sigma'$ are listed in Table 1. Clearly, the result holds for $i = 1$ and $i = 2$.

Insert Table 1 here

Now suppose that $i = m+1$ for $1 \leq m \leq \lfloor n/2 \rfloor$. The start times of job $2m+1$ in $\sigma$ and $\sigma'$ are given by

$$T'_{2m+1} = \max(T'_{2m} + S'_{2m}, T_{2m-1} + S_{2m-1} + p)$$
$$= \max(T_{2m-1} + S_{2m-1} + S_{2m}, T_{2m-2} + S_{2m-2} + S_{2m} + p, T_{2m-1} + S_{2m-1} + p)$$
$$= \max(T'_{2m-2} + S_{2m-2} + S_{2m} + p, T_{2m-1} + S_{2m-1} + p)$$
$$= T'_{2m-1} + S'_{2m-1} + p$$

by Lemma 8
If \( T_{2m+1} = T_{2m-2} + S_{2m-2} + S_{2m} + p \), then

\[
T_{2m+1} = T_{2m-2} + S_{2m-2} + S_{2m} + p \\
\leq T_{2m-1} + S_{2m} + p \quad \text{since } T_{2m-1} \geq T_{2m-2} + S_{2m-2} \\
\leq T'_{2m-1} + S_{2m} + p \quad \text{by assumption} \\
\leq T'_{2m-1} + S'_{2m-1} + p \quad \text{since } S'_{2m-1} = \max(S_{2m-1}, S_{2m}) \\
= T'_{2m+1}
\]

Otherwise, \( T_{2m+1} = T_{2m-1} + S_{2m-1} + p \leq T'_{2m-1} + S'_{2m-1} + p = T'_{2m+1} \). Consequently, \( T_{2i-1} \leq T'_{2i-1} \) for \( 1 \leq i \leq \lfloor n/2 \rfloor + 1 \).

If \( n \) is odd, then \( \lfloor n/2 \rfloor + 1 = (n+1)/2 \) and \( T_n \leq T'_n \). Since \( S_n = S'_n \) when \( n \) is odd,

\[
C_{\max}(\sigma) = T_n + S_n + p \leq T'_n + S'_n + p = C_{\max}(\sigma')
\]

If \( n \) is even, then

\[
C_{\max}(\sigma) = T_n + S_n + p \\
= \max(T_{n-1} + S_{n-1} + S_n + p, T_{n-2} + S_{n-2} + S_n + 2p)
\]

and

\[
C_{\max}(\sigma') = T'_n + S'_n + p \geq T'_{n-1} + S'_{n-1} + S'_n + p
\]

If \( C_{\max}(\sigma) = T_{n-1} + S_{n-1} + S_n + p \), then, obviously, \( C_{\max}(\sigma) \leq C_{\max}(\sigma') \).

Otherwise,

\[
C_{\max}(\sigma) = T_{n-2} + S_{n-2} + S_n + 2p \\
\leq T_{n-1} + S_n + 2p \leq T'_{n-1} + S_n + 2p \\
\leq T'_{n-1} + S'_{n-1} + S'_n + 2p \leq C_{\max}(\sigma') + p
\]

\( \Box \)

**Proposition 10** \( R^\infty_{LS} \leq 4/3 \) for the short setup problem.

**Proof.** From the proof of Lemma 9, it is clear that we may assume, without loss of generality, that \( n \) is even. Now
\[ C_{\text{max}}(\sigma) \leq C_{\text{max}}(\sigma') + p \quad \text{by Lemma 9} \]
\[ = T'_n + S'_n + p + p \]
\[ \leq T'_{n+1} + 2p \quad \text{since } T'_{n+1} \geq T'_n + S'_n \]
\[ = \sum_{i=1}^{n/2} a_{2i-1} + 2p \quad \text{by Lemma 8} \]

For an optimal solution,

\[ \text{OPT}(\sigma) = \text{OPT}(\sigma') \geq \frac{1}{2} \sum_{i=1}^{n} a'_i = \frac{1}{2} \left( \sum_{i=1}^{n/2} a'_{2i-1} + \sum_{i=1}^{n/2} a'_{2i} \right) \]

Since \( np \geq \sum_{i=1}^{n/2} a'_{2i-1} \geq \sum_{i=1}^{n/2} a'_{2i} \geq np/2 \), thus

\[ \frac{\sum_{i=1}^{n/2} a'_{2i}}{\sum_{i=1}^{n/2} a'_{2i-1}} \geq \frac{1}{2} \]

So

\[
\frac{C_{\text{max}}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{C_{\text{max}}(\sigma') + p}{\text{OPT}(\sigma')} \leq \frac{2(\sum_{i=1}^{n/2} a'_{2i-1} + 2p)}{\sum_{i=1}^{n/2} a'_{2i-1} + \sum_{i=1}^{n/2} a'_{2i}}
\]
\[ = \frac{2 \sum_{i=1}^{n/2} a'_{2i-1}}{\sum_{i=1}^{n/2} a'_{2i-1} + \sum_{i=1}^{n/2} a'_{2i}} \quad \text{as } n \to \infty \]
\[ = \frac{2}{1 + \frac{\sum_{i=1}^{n/2} a'_{2i}}{\sum_{i=1}^{n/2} a'_{2i-1}}} \leq \frac{4}{3} \]

Hereby, the result follows. \( \square \)

Next we consider the general case of equal processing time problem.

**Proposition 11** For an equal processing time schedule, decrease the setup times of all long setup jobs to \( p_s = p \), respectively. The total server waiting time does not change, while the makespan is reduced by the sum of long setup times minus \( lp \), that is, \( \sum_{i \in L} (s_i - p) \).

**Proof.** Suppose job \( i \) is a long setup job. Before and after the reduction, the server waiting times are given by

\[ W_{i-1} = (p - S_i - W_{i-2})^+ = 0; \quad W_{i-1}^s = (p - p - W_{i-2})^+ = 0 \]
since $S_i > p$. The server waiting times do not change, but the total setup time decreases by $\sum_{i \in L} (s_i - p)$. Hence the result follows. □

**Proposition 12** $R^\infty_{LS} = 4/3$ for the equal processing time problem.

**Proof.** Assume that $n = l + r$ is even. Decrease the setup times of all long setup jobs to $p$ and denote the new schedule by $\sigma^*$. From the proof of Proposition 10, we have

$$C_{\text{max}}(\sigma^*) \leq C_{\text{max}}(\sigma^{s'}) + p \leq \sum_{i=1}^{(l+r)/2} a'_{2^{i-1}} + 2p$$

So, by Proposition 11

$$C_{\text{max}}(\sigma) = C_{\text{max}}(\sigma^*) + \sum_{i \in L} (s_i - p) \leq \sum_{i=1}^{(l+r)/2} a'_{2^{i-1}} + 2p + \sum_{i \in L} (s_i - p)$$

By Proposition 4 in Abdekhodae et al. [3], there is an optimal schedule with a single run of purely long setup jobs. So

$$\text{OPT}(\sigma) = C_{\text{max}}(L) + C_{\text{max}}(R) \geq \sum_{i \in L} s_i + \frac{1}{2} \sum_{i \in R} a_i$$

$$= \sum_{i \in L} s_i + \frac{1}{2} \sum_{i \in R} + \frac{rp}{2}$$

**Case (i):** $l > r$

$$C_{\text{max}}(\sigma) \leq \sum_{i=1}^{(l+r)/2} a'_{2^{i-1}} + 2p + \sum_{i \in L} (s_i - p) \leq (l + r)p + 2p + \sum_{i \in L} (s_i - p)$$

$$= (r + 2)p + \sum_{i \in L} s_i$$

$$\text{OPT}(\sigma) = \sum_{i \in L} s_i + \frac{1}{2} \sum_{i \in R} s_i + \frac{rp}{2} \geq \sum_{i \in L} s_i + \frac{rp}{2}$$

$$C_{\text{max}}(\sigma) \leq \frac{(r + 2)p + \sum_{i \in L} s_i}{\sum_{i \in L} s_i + \frac{rp}{2}} = 2 - \frac{1 - \frac{2p}{\sum_{i \in L} s_i}}{1 + \frac{rp}{2\sum_{i \in L} s_i}}$$

$$\rightarrow 2 - \frac{1}{1 + \frac{rp}{2\sum_{i \in L} s_i}} \text{ as } n \rightarrow \infty$$
Since $\sum_{i \in L} s_i \geq lp$ and $l > r$, 
\[
\frac{rp}{2 \sum_{i \in L} s_i} \leq \frac{rp}{2lp} < \frac{1}{2}
\]
Thus $C_{\text{max}}(\sigma)/\text{OPT}(\sigma) < 4/3$.

Case (ii): $l \leq r$

Recall that $L_1 = \{\text{job } i : s_i > p, i \text{ is processed on } M_1\}$ and $R_1 = \{\text{job } i : s_i \leq p, i \text{ is processed on } M_1\}$.

\[
C_{\text{max}}(\sigma) \leq \frac{(l+r)/2}{2} + \frac{\sum_{i \in L} s_i}{2} + \frac{\sum_{i \in R} s_i}{2} 
= \sum_{i \in L_1} a_i + \sum_{i \in R_1} a_i + 2p + \sum_{i \in L} (s_i - p) 
\leq (2l_1 - l + r_1 + 2)p + \sum_{i \in R_1} s_i + \sum_{i \in L} s_i 
= (l_1 - l)p + \frac{(l + r + 4)p}{2} + \sum_{i \in R_1} s_i + \sum_{i \in L} s_i 
\leq \frac{(l + r + 4)p}{2} + \sum_{i \in R_1} s_i + \sum_{i \in L} s_i
\]
since $l_1 + r_1 = (l + r)/2$.

\[
\text{OPT}(\sigma) \geq \sum_{i \in L} s_i + \frac{1}{2} \sum_{i \in R} s_i + \frac{rp}{2} \geq \sum_{i \in L} s_i + \frac{1}{2} \sum_{i \in R} s_i + \frac{rp}{2}
\]

The case in which $l + r$ is odd can be solved similarly. Denote $\sum_{i \in L} s_i$ and $\sum_{i \in R_1} s_i$ by $\alpha$ and $\beta$, respectively.

\[
\frac{C_{\text{max}}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{2\alpha + 2\beta + (l + r + 4)p}{2\alpha + \beta + rp}
\]

$2\alpha + \beta + rp \to \infty$ and $\frac{4p}{2\alpha + \beta + rp} \to 0$, as $n \to \infty$. So

\[
\frac{C_{\text{max}}(\sigma)}{\text{OPT}(\sigma)} \leq 1 + \frac{\beta + lp}{2\alpha + \beta + rp} \leq 1 + \frac{\beta + lp}{2lp + \beta + rp}
\]
since $\alpha \geq lp$

\[
= 2 - \frac{lp + rp}{2lp + \beta + rp} \leq 2 - \frac{lp + rp}{2lp + rp}
\]

16
\[1 + \frac{1}{2 + r/l} \leq \frac{4}{3}\] since \(r/l \geq 1\)

When \(\beta = 0\) and \(l = r\), the ratio is maximized. Hence \(R_{LS}^\infty \leq \frac{4}{3}\).

To demonstrate that the bound is tight, consider the following instance: there are \(n\) jobs of which \(s_{2i-1} = p\) and \(s_{2i} = \epsilon\) for \(1 \leq i \leq n/2\), and \(n\) is divisible by 4. The makespan of list scheduling is essentially \(np\). An optimal solution has an \(\epsilon\) setup time job scheduled first, following by all \(p\) setup time jobs, and the remaining \(\epsilon\) setup time jobs processed last. Now let \(\epsilon \to 0\). The latest completion time of the optimal solution is \(3np/4\). Thus \(C_{max}(LS)/OPT = 4/3\), which shows that the bound is best possible. \(\square\)

4.2 New Heuristic

According to Sgall [13], for a one by one job scheduling model, an on-line algorithm needs to assign each job of some sequence to a particular machine and time slot. However, in order to obtain better asymptotic performance, and under the general assumption in this paper that all jobs are processed alternately on two machines, we propose a definition of weak on-line algorithms, one that just specifies the positions of each job in a schedule. The final starting time of a job in position \(i\) is fixed once we know which jobs are in position \(1, 2, \cdots i - 1\).

Let us define two subsets, \(G\) and \(H\), before introducing the new heuristic. \(G = \{\text{job } i : s_i > p/2\}\) and \(H = \{\text{job } i : s_i \leq p/2\}\), while \(g\) and \(h\) indicate the numbers of jobs in subsets \(G\) and \(H\), respectively. Let a superscript \((')\) represent the same parameters in \(\sigma'\), where \(\sigma'\) is the schedule after the arrangement indicated in the previous subsection.

**Heuristic E**: Always assign \(M_1\) to current job \(i\) unless both of the following conditions are satisfied.

1. There is an idle place on \(M_2\) (for some \(2j\)), that is, no job has been assigned to the position, and
2. The job scheduled at position \(2j - 1\) and the current job \(i\) are both from \(G\) or both from \(H\).

If both of the conditions are satisfied, then schedule job \(i\) in position \(2j\).
Example:

An instance of equal processing time jobs:
\[ s_1 = 0.6p \ (G), s_2 = 0.3p \ (H), s_3 = 0.2p \ (H), s_4 = 0.7p \ (G), s_5 = 0.8p \ (G), s_6 = 0.6p \ (G), s_7 = 0.2p \ (H), s_8 = 0.9p \ (G). \]

The schedule of \( E \) have all jobs processed in the sequence:
1,4,2,3,5,6,7,8 \((GGHGGH \ G)\), where jobs 4,3 and 6 are processed on \( M_2 \).

Observations:

We make the following observations about an arbitrary \( E \) schedule \( \sigma \).

(1) If \( g \) (\( h \)) is odd and \( h \) (\( g \)) is even, there is one and only one idle position \( 2j \) in \( \sigma \) for some \( j \), while the job scheduled in position \( 2j - 1 \) is from \( G \) (or \( H \)).
(2) If \( g \) and \( h \) are both even, there are no idle positions.
(3) If \( g \) and \( h \) are both odd, there are two idle positions, one of which is at the end of \( \sigma \).

From the observations above, there is at most one idle position, excluding the last one, in an \( E \) schedule. Suppose there is an idle position \( 2j \) in a schedule \( \delta \) and its neighbor job \( 2j - 1 \) is from subset \( G \) (or \( H \)). Add an extra job \( d \) with setup time of \( p/2 + \epsilon \) (or \( p/2 \)), where positive number \( \epsilon \rightarrow 0 \). Then the scheduler will assign it to position \( 2j \). Let \( \delta_d \) be the schedule with the extra job.

\[
C_{\text{max}}(\delta) \leq C_{\text{max}}(\delta_d)
\]

\[
OPT(\delta) \geq \begin{cases} 
OPT(\delta_d) - 3p/2 - \epsilon & \text{job } 2j - 1 \in G \\
OPT(\delta_d) - 3p/2 & \text{job } 2j - 1 \in H 
\end{cases}
\]

So

\[
\frac{C_{\text{max}}(\delta)}{OPT(\delta)} \leq \begin{cases} 
\frac{C_{\text{max}}(\delta_d)}{OPT(\delta_d) - 3p/2 - \epsilon} & \text{job } 2j - 1 \in G \\
\frac{C_{\text{max}}(\delta_d)}{OPT(\delta_d) - 3p/2} & \text{job } 2j - 1 \in H 
\end{cases}
\]
\[
\frac{C_{\max}(\delta_d)}{OPT(\delta_d)} \quad \text{as } n \to \infty
\]

Similarly, if there is an idle position at the end of a schedule, a similar argument shows that this has an insignificant impact on the asymptotic competitive performance.

So to find the asymptotic competitive ratio of \(E\), we may assume that \(g\) and \(h\) are both even, that is, there is no idle position in an \(E\) schedule. Under this assumption, \(E\) schedules have similar properties to those of LS schedules, except that the difference between the setup times of two neighborhood jobs (job \(2j - 1\) and job \(2j\)) is at most \(p/2\) for an \(E\) schedule, while it is at most \(p\) for an LS schedule. These properties together with Lemmas 8 and 9 yield the following proposition.

**Proposition 13** \(R_E^\infty = 6/5\) for the short setup problem.

**Proof.** Let \(G'_1\) and \(G'_2\) (\(H'_1\) and \(H'_2\)) denote the subsets of jobs in \(\sigma'\), in which \(G\) (\(H\)) jobs are processed by \(M_1\) and \(M_2\), respectively.

\[
C_{\max}(\sigma) \leq C_{\max}(\sigma') + p = T'_n + S'_n + p + p
\]

\[
\leq T'_{n+1} + 2p \quad \text{since } T'_{n+1} \geq T'_n + S'_n
\]

\[
= \sum_{i=1}^{n/2} a'_{2i-1} + 2p \quad \text{by Lemma 8}
\]

\[
= \sum_{i \in G'_1} a'_i + \sum_{i \in H'_1} a'_i + 2p = \sum_{i \in G'_2} a'_i + \sum_{i \in H'_2} a'_i
\]

\[
OPT(\sigma) = OPT(\sigma') \geq \frac{1}{2} \sum_{i=1}^{n} a'_i
\]

\[
= \frac{1}{2} \left( \sum_{i \in G'_1} a'_i + \sum_{i \in G'_2} a'_i + \sum_{i \in H'_1} a'_i + \sum_{i \in H'_2} a'_i \right)
\]

Let \(\alpha = \sum_{i \in G'_1} a'_i + \sum_{i \in H'_1} a'_i\) and \(\beta = \sum_{i \in G'_2} a'_i + \sum_{i \in H'_2} a'_i\). Since \(S'_i > p/2\) when job \(i \in G'_1 \cup G'_2\), and \(S'_i \leq p/2\) when job \(i \in H'_1 \cup H'_2\),

\[
\frac{g}{2} \times \frac{3p}{2} + \frac{h}{2} \times p \leq \beta \leq \alpha \leq \frac{g}{2} \times 2p + \frac{h}{2} \times \frac{3p}{2}
\]

Besides, we have \(n = g + h\). So
\[\frac{\beta}{\alpha} > \frac{3g + \frac{b}{2}}{4g + \frac{3b}{4}} = \frac{3g + 2h}{4g + 3h} = \frac{3(n-h) + 2h}{4(n-h) + 3h} = \frac{3n-h}{4n-h}\]

Since \(\frac{\partial}{\partial h} \frac{3n-h}{4n-h} = \frac{-n}{(3n-h)^2} < 0\), \(\beta/\alpha\) achieves its minimum of 2/3, when \(h = n\).

\[
\frac{C_{max}(\sigma)}{OPT(\sigma)} \leq \frac{C_{max}(\sigma') + p}{OPT(\sigma')} \leq \frac{2(\alpha + 2p)}{\alpha + \beta} \to \frac{2\alpha}{\alpha + \beta} \quad \text{as } n \to \infty
\]

\[
= \frac{2}{1 + \beta/\alpha} \leq \frac{2}{1 + 2/3} = \frac{6}{5}
\]

The instance \(I\) of EPP, in which \(s_{2i-1} = p/2\) and \(s_{2i} = \epsilon\) for \(1 \leq i \leq n/2\) where \(n\) is divisible by 8, achieves the bound, since by an argument similar to that in Proposition 12 we have

\[
C_{max}(I) = \frac{n}{2} \times \frac{3}{2}p = \frac{3np}{4}
\]

\[
OPT(I) = \frac{1}{2} \sum_{i=0}^{n} a_i = \frac{1}{2} \left[ \frac{n}{2} (\frac{p}{2} + p) + p \right] = \frac{5np}{8}
\]

\[
\frac{C_{max}(I)}{OPT(I)} = \frac{3np/4}{5np/8} = \frac{6}{5}
\]

The conclusion follows. 

Proposition 11 is also applicable for \(E\) schedules. Furthermore, we can obtain the same upper bound of \(E\) for the general case of the equal processing time problem with the analogous process to that in Proposition 12. Therefore, \(R^\infty_E = 6/5\) for EPP.

5 **RESP** \((P_2, S_1 \mid p_i \leq a_j, \ s_i = s \ \forall i, j \mid C_{max})\)

For a regular set of jobs, \(p_i \leq a_j \ \forall i, j\). A necessary and sufficient condition for regularity for equal setup time jobs is that \(\max p_i - \min p_j \leq s\) where \(s_i = s \ \forall i\) [3], that is, \(p_i \leq v + s \ \forall i\). As for the previous special case Abdekhodaei et al. [3] showed that the off-line version of this problem is NP-hard. We now introduce the following definitions. \(v = \min p_i, \ L = \{\text{job } i : \ p_i > s\}\) and \(R = \{\text{job } i : \ p_i \leq s\}\). Let the machine number which processes the jobs in a subset be represented by an index and let lower cases represent the numbers of jobs in each subset. Then, \(n = l + r = l_1 + l_2 + r\).
5.1 LS

We rearrange an arbitrary schedule \( \sigma \) of equal setup time jobs in following way: If \( P_{2i-1} < P_{2i} \), swap jobs \( 2i - 1 \) and \( 2i \) for \( 1 \leq i \leq \lfloor (n-1)/2 \rfloor \).

Denote the new schedule after the rearrangement by \( \sigma' \) and the processing time of the \( i \)th job of \( \sigma' \) by \( P'_{2i} \). Clearly, \( P'_{2i-1} \geq P'_{2i} \) for \( 1 \leq i \leq \lfloor (n-1)/2 \rfloor \).

**Lemma 14** \( T'_{2i-1} = \sum_{j=1}^{i-1} \left[ 2s + \left( P'_{2j-1} - s \right)^+ \right] \) and \( T'_{2i} = T'_{2i-1} + s \), where \( 2 \leq i \leq \lfloor (n+1)/2 \rfloor \) for \( \sigma' \).

**Proof.** The proof is by induction. For \( i = 2 \), we have that

\[
\begin{align*}
T'_1 &= 0 \\
T'_2 &= s \\
T'_3 &= \max(s + s, s + P'_1) = 2s + (P'_1 - s)^+ \\
T'_4 &= \max(T'_3 + s, T'_2 + s + P'_2) = \max(3s, 2s + P'_1, 2s + P'_2) \\
&= \max(3s, 2s + P'_1) \quad \text{since } P'_1 \geq P'_2 = T'_3 + s
\end{align*}
\]

So the result is valid for \( i = 2 \). For \( i = m \), assume that

\[
\begin{align*}
T'_{2m-1} &= \sum_{j=1}^{m-1} \left[ 2s + \left( P'_{2j-1} - s \right)^+ \right] \\
T'_{2m} &= T'_{2m-1} + s
\end{align*}
\]

Consider the case that \( i = m + 1 \) for \( 2 \leq m \leq \lfloor (n-1)/2 \rfloor \). We have that

\[
\begin{align*}
T'_{2m+1} &= \max(T'_{2m} + s, T'_{2m-1} + s + P'_{2m-1}) \\
&= \max(T'_{2m-1} + s + s, T'_{2m-1} + s + P'_{2m-1}) \\
&= \max \left\{ \sum_{j=1}^{m-1} \left[ 2s + \left( P'_{2j-1} - s \right)^+ \right] + 2s, \\
&\quad \sum_{j=1}^{m-1} \left[ 2s + \left( P'_{2j-1} - s \right)^+ \right] + s + P'_{2m-1} \right\} \\
&= \sum_{j=1}^{m} \left[ 2s + \left( P'_{2j-1} - s \right)^+ \right]
\end{align*}
\]

and

\[
21
\]
\( T'_{2m+2} = \max(T'_{2m+1} + s, T'_{2m} + s + P'_{2m}) \)

\[ = \max(T'_{2m+1} + s, T'_{2m-1} + 2s + P'_{2m}) \]

Since \( T'_{2m+1} \geq T'_{2m-1} + s + P'_{2m-1} \geq T'_{2m-1} + s + P'_{2m} \), \( T'_{2m+2} = T'_{2m+1} + s \).

Hence, the result follows. \( \Box \)

**Lemma 15** \( C_{\max}(\sigma) \leq C_{\max}(\sigma') \)

**Proof.** First, we will demonstrate that \( T_{2i-1} \leq T'_{2i-1} \) and \( T_{2i} \leq T'_{2i} \) for \( 0 \leq i \leq \lfloor (n + 1)/2 \rfloor \). Once again the proof is by induction. For \( i = 1 \) and \( i = 2 \), the start times of the first four jobs of \( \sigma \) and \( \sigma' \) are listed in Table 2, respectively. So the result is true for \( i = 0 \) and \( i = 1 \), since \( P'_{1} = \max(P_{1}, P_{2}) \). Now suppose that \( i = m + 1 \) for \( 1 \leq m \leq \lfloor (n - 1)/2 \rfloor \). The start times of jobs \( 2m + 1 \) and \( 2m + 2 \) are given in Table 3.

**Insert Table 2 here**

**Insert Table 3 here**

By inductive assumption, \( T_{2m-1} \leq T'_{2m-1} \) and \( T_{2m} \leq T'_{2m} \). As \( P'_{2m-1} = \max(P_{2m-1}, P_{2m}) \geq P_{2m-1} \), it is obvious that \( T_{2m+1} \leq T'_{2m+1} \). As for job \( 2m + 2 \), if \( T_{2m+2} = T_{2m+1} + s \), then \( T'_{2m+2} = T'_{2m+1} + s \geq T_{2m+1} + s = T_{2m+2} \). If \( T_{2m+2} = T_{2m} + s + P_{2m} \), then

\[ T'_{2m+2} = T'_{2m+1} + s = \max(T'_{2m} + s, T'_{2m-1} + s + P'_{2m-1}) + s \]

\[ \geq T'_{2m-1} + P'_{2m-1} + 2s \geq T'_{2m-1} + P_{2m} + 2s \quad \text{since} \quad P'_{2m-1} \geq P_{2m} \]

\[ = T'_{2m} + P_{2m} + s \quad \text{by assumption} \]

\[ \geq T_{2m} + P_{2m} + s \quad \text{since} \quad T'_{2m} = T'_{2m-1} + s \quad \text{by Lemma 14} \]

So we conclude that \( T_{2i-1} \leq T'_{2i-1} \) and \( T_{2i} \leq T'_{2i} \) for \( 1 \leq i \leq \lfloor (n + 1)/2 \rfloor \).

Since job \( n \) remains in place after the rearrangement, that is, \( P_{n} = P'_{n} \), we have

\[ C_{\max}(\sigma) = T_{n} + s + P_{n} \]

\[ C_{\max}(\sigma') = T'_{n} + s + P'_{n} = T'_{n} + s + P_{n} \]

Note that, if \( n \) is even and \( i = n/2 \), \( T_{2i} \leq T'_{2i} \), that is, \( T_{n} \leq T'_{n} \). On the
other hand if \( n \) is odd and \( i = \frac{n+1}{2} \), \( T_{2i-1} \leq T'_{2i-1} \), that is, \( T_n \leq T'_{n} \). Consequently, \( C_{\text{max}}(\sigma) \leq C_{\text{max}}(\sigma') \). \( \square \)

**Proposition 16** \( R_{LS}^\infty = 6/5 \) for the regular equal setup time problem.

**Proof.** The makespan of \( \sigma \) is bounded above thus,

\[
C_{\text{max}}(\sigma) \leq C_{\text{max}}(\sigma') = T'_n + s + P'_n
\]

\[
= \begin{cases} 
\sum_{i=1}^{n/2-1} \left[ 2s + (P'_{2i-1} - s) \right] + 2s + P'_n & \text{if } n \text{ is even} \\
\sum_{i=1}^{(n-1)/2} \left[ 2s + (P'_{2i-1} - s) \right] + s + P'_n & \text{if } n \text{ is odd}
\end{cases}
\]

**Case (i):** \( v \geq s \)

Then

\[
C_{\text{max}}(\sigma) \leq \begin{cases} 
\sum_{i=1}^{n/2-1} (s + P'_{2i-1}) + 2s + P'_n & \text{if } n \text{ is even} \\
\sum_{i=1}^{(n+1)/2} (s + P'_{2i-1}) & \text{if } n \text{ is odd}
\end{cases}
\]

For the optimal solution,

\[
\text{OPT}(\sigma) = \text{OPT}(\sigma') \geq \frac{1}{2} \sum_{i=1}^{n} a'_i = \frac{1}{2} \sum_{i=1}^{n} (s + P'_i)
\]

\[
= \begin{cases} 
\frac{1}{2} \left[ \sum_{i=1}^{n/2} (s + P'_{2i}) + \sum_{i=1}^{n/2} (s + P'_{2i-1}) \right] & \text{if } n \text{ is even} \\
\frac{1}{2} \left[ \sum_{i=1}^{(n-1)/2} (s + P'_{2i}) + \sum_{i=1}^{(n+1)/2} (s + P'_{2i-1}) \right] & \text{if } n \text{ is odd}
\end{cases}
\]

From the result above, and from the fact that we are considering the asymptotic competitive ratio, we may assume that \( n \) is even. Then

\[
\frac{C_{\text{max}}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{C_{\text{max}}(\sigma')}{\text{OPT}(\sigma')}
\]

\[
\leq \frac{2 \sum_{i=1}^{n/2-1} (s + P'_{2i-1}) + 4s + 2P'_n}{\sum_{i=1}^{n/2-1} (s + P'_{2i}) + \sum_{i=1}^{n/2-1} (s + P'_{2i-1}) + 2s + P'_n + P'_{n-1}}
\]
Denote $\sum_{i=1}^{n/2-1}(s + P'_{2i-1}) + s + P_n'$ and $\sum_{i=1}^{n/2-1}(s + P'_{2i}) + s + P'_{n-1}$ by $\alpha$ and $\beta$, respectively. Then the asymptotic competitive ratio is given by

$$C_{\text{max}}(\sigma) \leq \frac{2\alpha + 2s}{\alpha + \beta} = \frac{2 + 2s/\alpha}{1 + \beta/\alpha}$$

As $\alpha \to \infty$, $2s/\alpha$ is neglected. Since $v \leq P'_i \leq s + v \forall i$,

$$\frac{(s + v)n}{2} \leq \beta \leq \frac{(2s + v)n}{2} \text{ and } \frac{(s + v)n}{2} \leq \alpha \leq \frac{(2s + v)n}{2}$$

Thus, $\frac{\beta}{\alpha} \geq s + \frac{v}{s + v}$. Since $\frac{\partial(s + v)/(2s + v)}{\partial v} > 0$, $s + v/2s + v$ is minimized when $v$ achieves its minimal value $s$. Therefore, $\beta/\alpha \geq 2/3$. The competitive ratio is maximized when $\beta/\alpha$ achieves the minimum. So $C_{\text{max}}(\sigma)/OPT(\sigma) \leq 6/5$.

**Case (ii): $0 \leq v < s$**

Assume that $n$ is even. By Lemma 14,

$$C_{\text{max}}(\sigma') = \sum_{i=1}^{n/2-1} [2s + (P'_{2i-1} - s)^+] + 2s + P_n'$$

$$= \sum_{i=1}^{n/2-1} (P'_{2i-1} - s)^+ + (n/2 - 1)2s + 2s + P_n'$$

$$= \sum_{i \in L_1} (P'_i - s) + ns + P'_n$$

$$= \sum_{i \in L_1} P'_i + (n - l_1)s + P'_n \quad (2)$$

By Proposition 5 in [3], there exists an optimal solution in which all long jobs form a single run. The makespan of an optimal solution consists of two parts: the makespans of long jobs and short jobs, respectively. For the short jobs, since $W'_i = (P'_i - S'_{i+1} - W'_{i-1})^+ = 0$, $C_{\text{max}}(R) \geq \sum_{i \in k} s'_i + v$. As for the long jobs, their makespan lower bound is $\frac{1}{2} \sum_{i \in L_1 \cup L_2} a'_i$. Hence, we have that

$$OPT(\sigma') = C_{\text{max}}(L) + C_{\text{max}}(R) \geq \frac{1}{2} \sum_{i \in L_1 \cup L_2} a'_i + \sum_{i \in R} s'_i + v$$

$$= \frac{1}{2} l_1 + \frac{1}{2} l_2 + r) + \frac{1}{2} \sum_{i \in L_1} P'_i + \frac{1}{2} \sum_{i \in L_2} P'_i + v$$

24
\[ \geq \left( \frac{1}{2} l_1 + l_2 + r \right) s + \frac{1}{2} \sum_{i \in L_1} P'_i + v \quad \text{since } P'_i \geq s, \; i \in L_2 \]

\[ = (n - \frac{1}{2} l_1) s + \frac{1}{2} \sum_{i \in L_1} P'_i + v \quad \text{since } l_1 + l_2 + r = n \]

So

\[
\frac{C_{\text{max}}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{C_{\text{max}}'(\sigma')}{\text{OPT}(\sigma')} \leq \frac{\sum_{i \in L_1} P'_i + (n - l_1) s + P'_n}{(n - \frac{1}{2} l_1) s + \frac{1}{2} \sum_{i \in L_1} P'_i + v}
\]

\[
= 2 \frac{\sum_{i \in L_1} P'_i + 2(n - l_1) s + 2P'_n}{(2n - l_1) s + \sum_{i \in L_1} P'_i + 2v}
\]

\[
= 2 - \frac{2ns - 2P'_n + 4v}{(2n - l_1) s + \sum_{i \in L_1} P'_i + 2v}
\]

By assumption, \( l_1 \leq n/2 \) and \( l_1 s \leq \sum_{i \in L_1} P'_i \leq l_1(s + v) \). Thus,

\[
\frac{C_{\text{max}}(\sigma)}{\text{OPT}(\sigma)} \leq 2 - \frac{2ns - 2P'_n + 4v}{(2n - l_1) s + l_1(s + v) + 2v} = 2 - \frac{2ns - 2P'_n + 4v}{2ns + l_1 v + 2v}
\]

\[
\leq 2 - \frac{2ns - 2P'_n + 4v}{2ns + nv/2 + 2v}
\] (3)

To find the maximum of \( C_{\text{max}}(\sigma)/\text{OPT}(\sigma) \), consider

\[
\frac{\partial^2 (2ns - 2P'_n + 4v)}{\partial v^2} = \frac{4(2ns + nv/2 + 2v) - (n/2 + 2)(2ns - 2P'_n + 4v)}{(2ns + nv/2 + 2v)^2}
\]

\[
= \frac{4ns - n^2 s + nP'_n + 4P'_n}{(2ns + nv/2 + 2v)^2} < 0
\]

for sufficiently large \( n \), since \( P'_n < 2s \). Consequently, the ratio is maximized when \( v \) achieves its maximum. Hence, for \( 0 < v < s \),

\[
\frac{C_{\text{max}}(\sigma)}{\text{OPT}(\sigma)} \leq 2 - \frac{2ns - 2P'_n + 4s}{5ns/2 + 2s} = \frac{6}{5} \quad \text{as } n \rightarrow \infty
\]

The upper bound obtained in Proposition 16 is achieved by the sequence in which \( \min \; p_1 = s, \; p_{2i-1} = 2s \) and \( p_{2i} = s \) for \( 1 \leq i \leq n/2 \), where \( n \) is
divisible by 4. The makespan of list scheduling is $3ns/2$. An optimal solution has jobs processed in the following order: $p_1p_3 \cdots p_{n-3}p_{n-1}p_2p_4 \cdots p_{n-2}p_n$ and the optimal makespan is $(s + 2s)n/4 + 2sn/4 + s = 5ns/4 + s$. Thus $C_{\text{max}}(LS)/\text{OPT} \rightarrow 6/5$ when $n \rightarrow \infty$. □

5.2 New Heuristic

We slightly modify some definitions for heuristic EP to obtain the following heuristic for RESP.

Define the following job subsets:

$G = \{ \text{job } i : p_i > v + s/2 \}$ and $H = \{ \text{job } i : p_i \leq v + s/2 \}$, where $v = \min_i p_i$. $H_1^* = H_1 \cap L_1$; $H_2^* = H_2 \cap L_2$.

**Heuristic F:** Always assign $M_1$ to current job $i$ unless both of the following conditions are satisfied.

1. There is an idle place on $M_2$ (for some $2j$), that is, no job has been assigned to the position, and
2. The job scheduled at position $2j - 1$ and the current job $i$ are both from $G$ or both from $H$.

If both conditions are satisfied, then schedule the job $i$ in position $2j$.

**Observations:**

The observations made earlier about $E$ also apply to $F$ schedules. So we may assume that $g$ and $h$ are both even. By utilizing the properties of $F$ schedules, and applying Lemmas 14 and 15 we derive the following proposition.

**Proposition 17** $R_F^\infty = 10/9$ for the regular equal setup problem.

**Proof.** In terms of the values of $v$, there are three cases to consider.

**Case (i):** $v \geq s$

Inequality (1) holds here as well, that is,
\[
\frac{C_{\text{max}}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{C_{\text{max}}(\sigma')}{\text{OPT}(\sigma')}
\]
\[
\leq \frac{2 \sum_{i=1}^{n/2-1} (s + P'_{2i-1}) + 4s + 2P'_n}{\sum_{i=1}^{n/2-1} (s + P'_{2i-1}) + \sum_{i=1}^{n/2-1} (s + P'_{2i-1}) + 2s + P'_n + P'_{n-1}}
\]
\[
= \frac{2 + 2s/\alpha}{1 + \beta/\alpha} \rightarrow \frac{2}{1 + \beta/\alpha} \quad n \to \infty
\]

where \(\alpha = \sum_{i=1}^{n/2-1} (s + P'_{2i-1}) + s + P'_n\) and \(\beta = \sum_{i=1}^{n/2-1} (s + P'_{2i-1}) + s + P'_{n-1}\).

Since
\[
\frac{g}{2} \left(3s + v\right) + \frac{h}{2} (s + v) \leq \beta \leq \alpha \leq \frac{g}{2} (2s + v) + \frac{h}{2} \left(3s + v\right)
\]
and \(g + h = n\),

\[
\frac{3ns + 2nv - sh}{4ns + 2vn - sh}
\]

Let \(x = 2nv - sh\) and \(\frac{\partial^2 ns + x}{\partial x^2} = \frac{ns}{(4ns + x)^2} > 0\). So \(\frac{3ns + 2nv - sh}{4ns + 2nv - sh}\) is minimized when \(x\) is minimized. As \(v \geq s\) and \(h \leq n\), \(\min\{x\} = ns\) when \(v = s\) and \(h = n\). Hence \(C_{\text{max}}(\sigma)/\text{OPT}(\sigma) \leq 10/9\).

**Case (ii):** \(v \leq s/2\)

By inequality (3), that is,
\[
\frac{C_{\text{max}}(\sigma)}{\text{OPT}(\sigma)} \leq 2 - \frac{2ns - 2P'_n + 4v}{2ns + nv/2 + 2v}
\]
we have that

\[
\frac{C_{\text{max}}(\sigma)}{\text{OPT}(\sigma)} \leq 2 - \frac{2ns - 2P'_n + 4v}{2ns + ns/4 + 2v} \rightarrow \frac{10}{9} \quad \text{as } n \to \infty
\]

**Case (iii):** \(s/2 \leq v \leq s\)

We recall the definitions of sets \(L\), \(G\) and \(H\). Thus, \(p_i > s\) for \(i \in L\), \(p_i > v + s/2\) for \(i \in G\), \(p_i \leq v + s/2\) for \(i \in H\) and \(H^* = L \cap H\). Since \(v \geq s/2\), all
jobs in $G$ are long jobs, that is, $G \subseteq L$. Thus, $L_1 = G_1 \cup H_1^\ast$, $L_2 = G_2 \cup H_2^\ast$ and $l_1 + l_2 \geq g$. From equality (2),

$$C_{\max}(\sigma^{'}) = \sum_{i \in L_1} P'_i + (n - l_1) + P'_n$$

$$OPT(\sigma^{'}) \geq \left( \frac{1}{2} l_1 + \frac{1}{2} l_2 + r \right) s + \frac{1}{2} \sum_{i \in L_1} P'_i + \frac{1}{2} \sum_{i \in L_2} P'_i + v$$

$$\geq \left( \frac{1}{2} l_1 + \frac{1}{2} l_2 + r \right) s + \frac{1}{2} \sum_{i \in L_1} P'_i + \frac{1}{2} \sum_{i \in G_2 \cup H_2^*} P'_i + v$$

$$\geq \left( \frac{1}{2} l_1 + \frac{1}{2} l_2 + r \right) s + \frac{1}{2} \sum_{i \in L_1} P'_i + \frac{g}{4} \left( v + \frac{s}{2} \right) + \frac{1}{2} \sum_{i \in H_2^*} P'_i + v$$

$$\geq \left( \frac{1}{2} l_1 + \frac{1}{2} l_2 + r \right) s + \frac{1}{2} \sum_{i \in L_1} P'_i + \frac{g}{4} \left( v + \frac{s}{2} \right) + \frac{s}{2} (l_2 - g) + v$$

$$= \left( n - \frac{l_1}{2} \right) s + \frac{1}{2} \sum_{i \in L_1} P'_i + \frac{g}{4} \left( v - \frac{s}{2} \right) + v$$

Then

$$\frac{C_{\max}(\sigma)}{OPT(\sigma)} \leq \frac{\sum_{i \in L_1} P'_i + (n - l_1) s + P'_n}{OPT(\sigma')} \leq \frac{\sum_{i \in L_1} P'_i + (n - l_1) s + P'_n}{\left( n - \frac{l_1}{2} \right) s + \frac{1}{2} \sum_{i \in L_1} P'_i + \frac{g}{4} \left( v - \frac{s}{2} \right) + v}$$

$$= 2 - \frac{2ns - 2P'_n + g(v - \frac{s}{2}) + 4v}{(2n - l_1)s + \sum_{i \in L_1} P'_i + \frac{g}{2}(v - \frac{s}{2}) + 2v}$$

Since

$$\sum_{i \in L_1} P'_i \leq \frac{g}{2} (v + s) + \left( l_1 - \frac{g}{2} \right) \left( v + \frac{s}{2} \right) \quad \text{as } p_i \leq v + s \ \forall i$$

we have that

$$\frac{C_{\max}(\sigma)}{OPT(\sigma)} \leq 2 - \frac{2ns - 2P'_n + g(v - \frac{s}{2}) + 4v}{(2n - l_1)s + \frac{g}{2}(v + s) + \left( l_1 - \frac{g}{2} \right) \left( v + \frac{s}{2} \right) + \frac{g}{2}(v - \frac{s}{2}) + 2v}$$

$$= 2 - \frac{2ns + g(v - \frac{gs}{2}) + 4v - 2P'_n}{2ns + l_1 v - \frac{1}{2} s + \frac{g}{2} v + 2v}$$

$$= 2 - \frac{2ns + g(v - \frac{gs}{2})}{2ns + l_1 v - \frac{1}{2} s + \frac{g}{2} v + 2v} \quad \text{as } n \to \infty$$

$$\leq 2 - \frac{8ns + 4gv - 2gs}{7ns + 2nv + 2gv} \quad \text{since } l_1 \leq n/2$$

$$= \frac{6ns + 4nv + 2gs}{7ns + 2nv + 2gv}$$

28
\[ \frac{\partial \left( \frac{6ns + 4nv + 2gs}{7ns + 2nv + 2gv} \right)}{\partial g} = \frac{2s(7ns + 2nv + 2gv) - 2v(6ns + 4nv + 2gs)}{(7ns + 2nv + 2gv)^2} \]

\[ = \frac{14ns^2 - 8nv - 8nv^2}{(7ns + 2nv + 4gv)^2} \]

\[ = 0 \quad \text{when } v = (\sqrt{2} - 1/2)s \]

\[ \frac{\partial \left( \frac{6ns + 4nv + 2gs}{7ns + 2nv + 2gv} \right)}{\partial v} = \frac{4n(7ns + 2nv + 2gv) - (2g + 2n)(6ns + 4nv + 2gs)}{(7ns + 2nv + 2gv)^2} \]

\[ = \frac{16n^2s - 4g^2s - 16ngs}{(7ns + 2nv + 2gv)^2} \]

\[ = 0 \quad \text{when } g = (2\sqrt{2} - 2)n \]

Since \( s/2 \leq v \leq s \) and \( 0 \leq g \leq n \), the possible extrema of the bounded region are listed in Table 4. However, maxima may only be achieved at the points with asterisk (*).

**Insert Table 4 here**

So \( \frac{6ns + 4nv + 2gs}{7ns + 2nv + 2gv} \) achieves its maximum \( 10/9 \) when \( v = s \), \( g = 0 \) or \( v = s/2 \), \( g = n \).

Consider the following instance: \( p_{2i-1} = 3/2s \) and \( p_{2i} = s \) for \( 1 \leq i \leq n/2 \) and \( n \) is divisible by 4.

\[ C_{\max}(I) = \frac{n}{2} \left( \frac{3s}{2} + s \right) + \frac{s}{2} \]

\[ OPT(I) = \frac{n}{4} (s + s) + \frac{n}{4} \left( \frac{3s}{2} + s \right) + s \]

\[ C_{\max}(I) \rightarrow \frac{5ns}{8} + \frac{s}{2} \rightarrow \frac{10}{9} \]

as \( n \rightarrow \infty \)

Hence the result follows. \( \square \)

## 6 Conclusions and future work

In this paper, we dealt with three special cases of the on-line scheduling of \( P_2, S_1 | s_i | C_{\max} \). We applied the simply on-line algorithm – List Scheduling to the cases and found the asymptotic competitive ratio of LS for each case. A lower bound for any on-line algorithm was provided for the equal length job problem. For the other two cases – equal processing and equal setup time
problems, we developed two simple but effective on-line heuristics, which were shown to beat LS in terms of asymptotic performance. Table 5 concludes these results. The future work may be dedicated to the tighter lower bounds for the problems, as well as some new heuristics, which lead to better competitive ratios. Additionally, the on-line scheduling study may be extended to the general case of the problem and the situation with an arbitrary number of machines.

Insert Table 5 here

References


[7] Nicholas G. Hall, Chris N. Potts, and Chelliah Srikandarajah. Parallel


Fig. 1. Move of job \( p + q + 1 \)

Fig. 2. Optimal solution for \( l \geq r \)
Fig. 3. Worst case of list scheduling and optimal schedule

Table 1

\[ i = 1 \text{ and } i = 2 \]

\begin{tabular}{ll}
\hline
\( \sigma \) & \( \sigma' \) \\
\hline
\( T_1 = 0 \) & \( T_1' = 0 \) \\
\( T_2 = S_1 \) & \( T_2' = S_1' = \max(S_1, S_2) \) \\
\( T_3 = \max(S_1 + S_2, S_1 + p) \) & \( T_3' = \max(S_1' + S_2', S_1' + p) \) \\
& \( = S_1 + p \) & \( = \max(S_1 + p, S_2 + p) \) \\
\hline
\end{tabular}
**Table 2**

$i = 1$ and $i = 2$

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>$\sigma'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1 = 0$</td>
<td>$T_1' = 0$</td>
</tr>
<tr>
<td>$T_2 = s$</td>
<td>$T_2' = s$</td>
</tr>
<tr>
<td>$T_3 = \max(s + s, s + P_1)$</td>
<td>$T_3' = \max(2s, s + P_1')$</td>
</tr>
<tr>
<td>$T_4 = \max(T_3 + s, T_2 + s + P_2)$</td>
<td>$T_4' = \max(T_3' + s, T_2' + s + P_2')$</td>
</tr>
<tr>
<td></td>
<td>$= \max(3s, 2s + P_1, 2s + P_2)$</td>
</tr>
</tbody>
</table>

**Table 3**

$i = m + 1$

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>$\sigma'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{2m+1} = \max(T_{2m} + s, T_{2m-1} + s + P_{2m-1})$</td>
<td>$T_{2m+1}' = \max(T_{2m} + s, T_{2m-1} + s + P_{2m-1}')$</td>
</tr>
<tr>
<td>$T_{2m+2} = \max(T_{2m+1} + s, T_{2m} + s + P_{2m})$</td>
<td>$T_{2m+2}' = \max(T_{2m+1}' + s, T_{2m}' + s + P_{2m}')$</td>
</tr>
<tr>
<td></td>
<td>$= T_{2m+1}' + s$ (by Lemma 14)</td>
</tr>
</tbody>
</table>
Table 4

Possible Extrema

<table>
<thead>
<tr>
<th>$v$</th>
<th>$g$</th>
<th>$\frac{6ns+4nr+2gs}{7ns+2nr+2gr}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\sqrt{2} - 1/2)s$</td>
<td>$(2\sqrt{2} - 2)n$</td>
<td>1.094</td>
</tr>
<tr>
<td>$s/2$</td>
<td>$n$</td>
<td>10/9</td>
</tr>
<tr>
<td>$s/2$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$s$</td>
<td>0</td>
<td>10/9</td>
</tr>
<tr>
<td>$s$</td>
<td>$n$</td>
<td>12/11</td>
</tr>
</tbody>
</table>

Table 5

Current asymptotic bounds for $P_2, S_1 | p_i \leq a_j \forall i, j | C_{max}$

<table>
<thead>
<tr>
<th>lower bound</th>
<th>LS</th>
<th>new heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELP</td>
<td>4/3</td>
<td>3/2</td>
</tr>
<tr>
<td>EPP</td>
<td>–</td>
<td>4/3</td>
</tr>
<tr>
<td>RESP</td>
<td>–</td>
<td>6/5</td>
</tr>
</tbody>
</table>
Title: On-line Scheduling of Two Parallel Machines with a Single Server

Abstract: In this paper, we consider the on-line scheduling of two parallel identical machines sharing a single server with the objective of minimizing the latest completion time of all jobs. Each job has to be setup by the server before being processed on one of the machines. Three special cases: equal length jobs, equal processing times and regular equal setup times are considered and the asymptotic competitive ratios of list scheduling are determined. Also, a lower bound for the equal length job case is given, and two heuristics with tight asymptotic competitive ratios for the other two cases are proposed.