Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 7, No. 3, pp. 551–564, September 2011

AN ANALYSIS OF LS ALGORITHM FOR THE PROBLEM OF SCHEDULING MULTIPLE JOBS ON MULTIPLE UNIFORM PROCESSORS WITH READY TIME*

Wei Ding and Yi Zhao

Abstract: In the paper we mainly study the C_{max} problem for scheduling n jobs on m uniform processors provided each job has a ready time. We first propose an LS algorithm based on uniform processors with ready time. We then obtain under this LS algorithm one tight bound of the ratio of the approximate solution T^{LS} to the optimal solution T^* for any $m \geq 3$ provided T^* is bigger than the processing time of the latest finish job. Moreover, we get under this LS algorithm an upper bound of the ratio of the approximate solution T^{LS} to the optimal solution T^* for any $m \geq 3$.

Key words: *Heuristic algorithm, LS algorithm, LPT algorithm, processors, tight bound*

Mathematics Subject Classification: *90B35, 68M20*

1 Introduction

The problem of scheduling n jobs $\{J_1, J_2, \cdots, J_n\}$ with given processing time on m uniform processors $\{M_1, M_2, \cdots, M_m\}$ with an objective of minimizing the makespan is one of the most well-studied problems in the scheduling literature, where processing J_j after J_i needs ready time $w(i,j)$. It has been proved to be $NP-hard$, cf. [10]. Therefore, the study of heuristic algorithms will be important and necessary for this scheduling problem. In fact, hundreds of scheduling theory analysts have cumulatively devoted an impressive number of papers to the worst-case and probabilistic analysis of numerous approximation algorithms for this scheduling problem.

In 1969 Graham [7] showed in his fundamental paper that the bound of this scheduling problem is $2 - \frac{1}{m}$ as $w(i,j) = 0$ under the LS (List Scheduling) algorithm and the tight bound is $\frac{4}{3} - \frac{1}{3m}$ under the LPT (Longest Processing Time) algorithm. In 1993 Ovacik and Uzsoy [9] proved the bound is $4 - \frac{2}{m}$ as $w(i,j) \leq t_j$, where t_j is the processing time of the job J_j, under the LS algorithm. In 2003 Imreh [8] studied the on-line and off-line problems on two groups of uniform processors, presented the LG (Load Greedy) algorithm, and showed that the bound about minimizing the makespan is $2+\frac{m-1}{k}$ and the bound about minimizing the sum of finish time is $2+\frac{m-2}{k}$, where m and k are the numbers of two groups of uniform processors. Gairing et al. (2007, [6]) proposed a simple combinatorial algorithm for the problem of scheduling n jobs on m uniform processors to minimize a cost stream and showed it is effective and of low complexity.

*This work was partially supported by (No. 10971234 and No. 10671213). The authors thank the referees for their valuable comments and suggestions.

Besides the above well-studied scheduling problem, one may face the problem of scheduling multi groups of jobs on multi processors in real production systems, such as, the problem of processing different types of yarns on spinning machines in spinning mills. Recently, the problems of scheduling multi groups of jobs on multi processors were studied provided each job has no ready time. In 2004 Ding [1] obtained a tight bound $T^{LPT}/T^* \leq 2$ for the problem of scheduling two groups of jobs on two special-purpose processors and m general-purpose processors under an LPT algorithm. In 2005 Ding [2] gave a bound $T^{LPT}/T^* \leq 4/3$ for the problem of scheduling three groups of jobs on three special-purpose processors and one general-purpose processor under an LPT algorithm. In the same year Ding [3] got a bound $T^{LPT}/T^* \leq 5/4$ for the problem of scheduling four groups of jobs on four special-purpose processors and one general-purpose processor under an LPT algorithm. In 2006 Ding [4] proposed a bound $T^{LPT}/T^* \leq (n+1)/n$ for the problem of scheduling n groups of jobs on one special-purpose processors and n general-purpose processors under an LPT algorithm. In 2008 Ding [5] presented a bound

$$\frac{T^{LPT}}{T^*} \leq \begin{cases} \frac{2m+1}{m+1}, & \text{if} \quad m \geq n-1, \\ \frac{m+n}{m+1}, & \text{if} \quad m < n-1, \end{cases}$$

for the problem of scheduling n groups of jobs on n special-purpose processors and m general-purpose processors under an LPT algorithm.

However, if each job has a ready time, then the problem of scheduling multi jobs on multi processors at different speeds has not been studied yet. Note that the LPT algorithm is not a effective way to deal with such a problem if each job has a ready time. Meanwhile, the classical LS algorithm is only useful to solve the problem of scheduling multi jobs on multi processors at same speeds. Therefore, our purpose of this study is to propose an LS algorithm based on uniform processors with ready time and to use this new algorithm to analyze this problem provided each job has a ready time and processors have different speeds.

The remainder of the paper is organized as follows. In Section 2, we proposed an LS algorithm for the problem of scheduling n jobs on m uniform processors provided each job has a ready time. In Section 3, we obtain under this LS algorithm one tight bound of the ratio of the approximate solution T^{LS} to the optimal solution T^* for any $m \geq 3$ provided T^* is bigger than the processing time of the latest finish job. Moreover, we get under this LS algorithm an upper bound for the ratio of the approximate solution T^{LS} to the optimal solution T^* for any $m \geq 3$.

Notation. As above and henceforth, we let J_i $(i = 1, 2, \cdots, n)$ denote the ith job and let M_i $(i = 1, 2, \cdots, m)$ denote the ith processor, respectively. We then denote by t_i $(i = 1, 2, \cdots, n)$ the processing time of J_i and by s_i $(i = 1, 2, \cdots, m)$ the speed of the processor M_i, respectively.

Set $s := \min_{1 \leq i \leq m} s_i$. Let $s_i^{'} = s_i/s$ $(i = 1, 2, \cdots, m)$ denote the relative speed of the processor M_k by comparing s_i with the smallest speed s. If no ambiguity, we still use s_i $(i = 1, 2, \cdots, m)$ to denote $s_i^{'}$. Thus, we may assume that the smallest speed s is equal to 1. In contrast to the smallest speed s, we have $s_i \geq 1$ $(i = 1, 2, \cdots, m)$.

If the job J_j $(j = 1, 2, \cdots, n)$ is processed after the job J_l $(l = 1, 2, \cdots, n)$, then we use $w(j,l)$ to denote the ready time – the time a processor spends waiting for reassignment when it could be running. Additionally, we let α denote the least upper bound of the ratio of the ready time $w(j,l)$ to the processing time t_l for $j, l = 1, 2, \cdots, n$, i.e.,

$$\alpha = \max_{j,\, l=1,2,\cdots,n} \Big\{ \frac{w(j,l)}{t_l} \Big\}.$$

If the job J_j is earlier than the job J_i to be assigned to a processor, then we write $J_j \prec J_i$. If the job J_i is assigned to the processor M_k, then we write $J_i \in M_k$. Let t_i/s_k denote the actual processing time of the job J_i on the processor M_k and let $ML_k(J_i)$ $(k = 1, 2, \cdots, m)$ denote the set of jobs assigned in the processor M_k before the job J_i is assigned, i.e.,

$$ML_k(J_i) = \{J_j | J_j \prec J_i, J_j \in M_k\}, \qquad k = 1, 2, \cdots, m.$$

Let $MT_k(J_i)/s_k$ $(k = 1, 2, \cdots, m)$ stand for the actual finish time of the processor M_k before the job J_i is assigned and

$$MT_k(J_i) = \sum_{J_j \in M_k, J_j \prec J_i} (w(*, j) + t_j), \qquad k = 1, 2, \cdots, m.$$

Next, we write T^{LS} as the actual latest finish time of m processors under an LS algorithm and T^* as the actual latest finish time of m processors under the optimal algorithm, respectively. We finally denote T^{LPT} by the approximate solution under an LPT algorithm, T^{LPT}/T^* by the bound of a scheduling problem under the LPT algorithm, and T^{LS}/T^* by the bound of a scheduling problem under the LS algorithm, respectively.

2 An *LS* algorithm

In the section, we will propose an LS algorithm for this scheduling problem.

The algorithm is defined by the fact that whenever a processor becomes idle for assignment, the first job unexecuted is taken from the list and assigned to this processor. If there are no less than one processor being idle, then the algorithm chooses the processor with the smallest index. In addition, there is an arbitrary order for the jobs at the beginning of being processed.

The steps of this LS algorithm are the following:

Step 1. Initialization.
Set $j = 1$, $ML_k(J_j) = \emptyset$, $MT_k(J_j) = 0$, $k = 1, 2, \cdots, m$.

Step 2. Choose the first idle processor.
Set $p = \min\{\, i \big| \, MT_i(J_j)/s_i = \min_{1 \le k \le m} MT_k(J_j)/s_k\}$.

Step 3. Update the assignment and the latest finish processor M_p.
If $j \le n$, then set $ML_p(J_{j+1}) = ML_p(J_j) + \{J_j\}$,

$$MT_p(J_{j+1}) = MT_p(J_j) + w(*, j) + t_j, \quad j = j + 1.$$

After that go to Step 2.

Step 4. If $j > n$ then set $T^{LS} = \max_{1 \le k \le m} \{MT_k/s_k\}$. Output the assignment of each processor ML_k $(k = 1, 2, \cdots, m)$ and the latest finish time T^{LS}.

3 Analysis of the *LS* algorithm

In the section, we first obtain under the LS algorithm one tight bound of the ratio of the approximate solution T^{LS} to the optimal solution T^* for any $m \ge 3$ provided T^* is bigger than the processing time of the latest finish job. Then, we get under the LS algorithm an upper bound of the ratio of the approximate solution T^{LS} to the optimal solution T^* for any $m \ge 3$.

Theorem 3.1. *Consider the problem of scheduling n jobs $\{J_1, J_2, \cdots, J_n\}$ on m uniform processors $\{M_1, M_2, \cdots, M_m\}$ provided each job has a ready time. Given the ready time $w(j,l)$ and the processing time t_l of J_l for $j, l = 1, 2, \cdots, n$, and let*

$$\alpha = \max_{j,\, l=1,2,\cdots,n} \{\frac{w(j,l)}{t_l}\}.$$

Assume that the optimal solution T^* is bigger than the processing time t_j of the latest finish job J_j. Then the tight bound of this scheduling problem under the LS algorithm is

$$\frac{T^{LS}}{T^*} \leq (1+\alpha)(1 + \frac{1}{s_k} - \frac{1}{\sum\limits_{i=1}^{m} s_i})$$

for any $m \geq 3$, where s_k is the speed of the latest finish processor.

Proof. Based on the LS algorithm introduced in Section 2, we may assume that some processor M_k $(1 \leq k \leq m)$ is the latest finish processor and the latest finish job is J_j $(1 \leq j \leq n)$. Then on the processor M_k, we have

$$T^{LS} = \frac{MT_k}{s_k}. \tag{3.1}$$

On other processors, we have

$$\frac{MT_i}{s_i} \geq \frac{MT_k - (w(*,j) + t_j)}{s_k}, \quad i = 1, 2, \cdots, m, \quad i \neq k. \tag{3.2}$$

Thus

$$\begin{aligned} \sum_{i=1}^{m} MT_i &= MT_k + \sum_{\substack{i=1 \\ i \neq k}}^{m} MT_i \\ &\geq s_k T^{LS} + \sum_{\substack{i=1 \\ i \neq k}}^{m} s_i T^{LS} - \frac{1}{s_k}(w(*,j) + t_j) \sum_{\substack{i=1 \\ i \neq k}}^{m} s_i \\ &= \sum_{i=1}^{m} s_i T^{LS} - \frac{1}{s_k}(w(*,j) + t_j) \sum_{\substack{i=1 \\ i \neq k}}^{m} s_i. \end{aligned} \tag{3.3}$$

On the other hand, by the assumption of the theorem, we have

$$T^* \geq t_j. \tag{3.4}$$

Since T^* is the optimal solution, it follows that

$$T^* \geq \frac{\sum\limits_{i=1}^{n} t_i}{\sum\limits_{i=1}^{m} s_i}. \tag{3.5}$$

By the definition of α and (3.4), we get

$$w(*,j) + t_j \leq (1+\alpha) t_j \leq (1+\alpha) T^*. \tag{3.6}$$

Then, by (3.3) and (3.6), we obtain

$$\sum_{i=1}^{m} MT_i \geq \sum_{i=1}^{m} s_i T^{LS} - \frac{1}{s_k}(1+\alpha)T^* \sum_{\substack{i=1\\ i\neq k}}^{m} s_i. \tag{3.7}$$

In view of the definition of α and (3.5), we deduce

$$\begin{aligned}\sum_{i=1}^{m} MT_i &= \sum_{i=1}^{m} \sum_{\{t_h\}\in ML_i} (w(*,h)+t_h) \\ &= \sum_{h=1}^{n}(w(*,h)+t_h) \\ &\leq (1+\alpha)\sum_{h=1}^{n} t_h \\ &\leq (1+\alpha)T^* \sum_{i=1}^{m} s_i. \end{aligned} \tag{3.8}$$

Using (3.7) and (3.8), we have

$$(1+\alpha)T^* \sum_{i=1}^{m} s_i \geq \sum_{i=1}^{m} MT_i \geq T^{LS} \sum_{i=1}^{m} s_i - \frac{1}{s_k}(1+\alpha)T^* \sum_{\substack{i=1\\ i\neq k}}^{m} s_i.$$

This yields

$$(1+\alpha)(\sum_{i=1}^{m} s_i + \frac{1}{s_k} \sum_{\substack{i=1\\ i\neq k}}^{m} s_i)T^* \geq T^{LS} \sum_{i=1}^{m} s_i.$$

Therefore

$$\begin{aligned}\frac{T^{LS}}{T^*} &\leq \frac{(1+\alpha)}{\sum\limits_{i=1}^{m} s_i}(\sum_{i=1}^{m} s_i + \frac{1}{s_k}\sum_{\substack{i=1\\ i\neq k}}^{m} s_i) \\ &= \frac{(1+\alpha)}{\sum\limits_{i=1}^{m} s_i}[\sum_{i=1}^{m} s_i + \frac{1}{s_k}(\sum_{i=1}^{m} s_i - s_k)] \\ &= (1+\alpha)(1+\frac{1}{s_k} - \frac{1}{\sum\limits_{i=1}^{m} s_i}). \end{aligned}$$

Next, the following examples will show the bound given in the theorem is tight for any $m \geq 3$.

Consider the following scheduling problems.

(1) As $m = 3$, we assume speeds of three processors M_1, M_2, M_3 are s_1, s_2, 1, respectively.

(i) If the processor M_1 is the latest finish processor, then we let the set of processors be $M = \{M_1, M_2, M_3\}$.

a) As $s_1 \leq s_2$, we set processing time and ready time of jobs are

Jobs J_i	J_1	J_2	J_3	J_4	J_5	J_6	J_7
Processing time t_i	s_1^2	s_2^2	s_1	$s_1 s_2$	s_2	$s_1 s_2$	s_1+s_2+1

and

Ready Time

$w(0,1) = \alpha s_1^2$	$w(1,4) = \alpha s_1 s_2$
$w(0,2) = \alpha s_2^2$	$w(2,6) = \alpha s_1 s_2$
$w(0,3) = \alpha s_1$	$w(3,5) = \alpha s_2$
$w(4,7) = \alpha(s_1 + s_2 + 1)$	$w(i,j) = 0$ for others i,j.

Then in this case, the LS schedule and the optimal schedule are

The LS Schedule

Processors	Jobs		
M_1	$t_1 = s_1^2$	$t_4 = s_1 s_2$	$t_7 = s_1 + s_2 + 1$
M_2	$t_2 = s_2^2$	$t_6 = s_1 s_2$	
M_3	$t_3 = s_1$	$t_5 = s_2$	

and

The Optimal Schedule

Processors	Jobs		
M_1	$t_4 = s_1 s_2$	$t_1 = s_1^2$	$t_3 = s_1$
M_2	$t_6 = s_1 s_2$	$t_2 = s_2^2$	$t_5 = s_2$
M_3	$t_7 = s_1 + s_2 + 1.$		

b) As $s_1 > s_2$, we set processing time and ready time of jobs are

Jobs J_i	J_1	J_2	J_3	J_4	J_5	J_6	J_7
Processing time t_i	s_1^2	s_2^2	s_1	$s_1 s_2$	$s_1 s_2$	s_2	$s_1 + s_2 + 1$

and

Ready Time

$w(0,1) = \alpha s_1^2$	$w(1,5) = \alpha s_1 s_2$
$w(0,2) = \alpha s_2^2$	$w(2,4) = \alpha s_1 s_2$
$w(0,3) = \alpha s_1$	$w(3,6) = \alpha s_2$
$w(5,7) = \alpha(s_1 + s_2 + 1)$	$w(i,j) = 0$ for others i,j.

Then in this case, the LS schedule and the optimal schedule are

The LS Schedule

Processors	Jobs		
M_1	$t_1 = s_1^2$	$t_5 = s_1 s_2$	$t_7 = s_1 + s_2 + 1$
M_2	$t_2 = s_2^2$	$t_4 = s_1 s_2$	
M_3	$t_3 = s_1$	$t_6 = s_2$	

and

The Optimal Schedule

Processors	Jobs		
M_1	$t_4 = s_1 s_2$	$t_1 = s_1^2$	$t_3 = s_1$
M_2	$t_5 = s_1 s_2$	$t_2 = s_2^2$	$t_6 = s_2$
M_3	$t_7 = s_1 + s_2 + 1.$		

Thus, if the processor M_1 is the latest finish processor, then we get

$$T^{LS} = \frac{MT_1}{s_1} = (1+\alpha)(s_1 + s_2 + \frac{s_1+s_2+1}{s_1}), \qquad T^* = s_1 + s_2 + 1,$$

and

$$\frac{T^{LS}}{T^*} = (1+\alpha)(1 + \frac{1}{s_1} - \frac{1}{s_1+s_2+1}).$$

(ii) If the processor M_2 is the latest finish processor, then we let the set of processors $M = \{M_2, M_1, M_3\}$.

a) As $s_1 < s_2$, we set processing time and ready time of jobs are

Jobs J_i	J_1	J_2	J_3	J_4	J_5	J_6	J_7
Processing time t_i	s_2^2	s_1^2	s_1	$s_1 s_2$	s_2	$s_1 s_2$	s_1+s_2+1

and

Ready Time

$w(0,1) = \alpha s_2^2$	$w(1,6) = \alpha s_1 s_2$
$w(0,2) = \alpha s_1^2$	$w(2,4) = \alpha s_1 s_2$
$w(0,3) = \alpha s_1$	$w(3,5) = \alpha s_2$
$w(6,7) = \alpha(s_1+s_2+1)$	$w(i,j) = 0$ for others i,j.

Then in this case, the LS schedule and the optimal schedule are

The LS Schedule

Processors	Jobs		
M_2	$t_1 = s_2^2$	$t_6 = s_1 s_2$	$t_7 = s_1+s_2+1$
M_1	$t_2 = s_1^2$	$t_4 = s_1 s_2$	
M_3	$t_3 = s_1$	$t_5 = s_2$	

and

The Optimal Schedule

Processors	Jobs		
M_1	$t_4 = s_1 s_2$	$t_2 = s_1^2$	$t_3 = s_1$
M_2	$t_6 = s_1 s_2$	$t_1 = s_2^2$	$t_5 = s_2$
M_3	$t_7 = s_1+s_2+1.$		

b) As $s_1 \geq s_2$, we set processing time and ready time of jobs are

Jobs J_i	J_1	J_2	J_3	J_4	J_5	J_6	J_7
Processing time t_i	s_2^2	s_1^2	s_1	$s_1 s_2$	$s_1 s_2$	s_2	s_1+s_2+1

and

Ready Time

$w(0,1) = \alpha s_2^2$	$w(1,4) = \alpha s_1 s_2$
$w(0,2) = \alpha s_1^2$	$w(2,5) = \alpha s_1 s_2$
$w(0,3) = \alpha s_1$	$w(3,6) = \alpha s_2$
$w(4,7) = \alpha(s_1+s_2+1)$	$w(i,j) = 0$ for others i,j.

Then in this case, the LS schedule and the optimal schedule are

The LS Schedule

Processors	Jobs		
M_2	$t_1 = s_2^2$	$t_4 = s_1 s_2$	$t_7 = s_1 + s_2 + 1$
M_1	$t_2 = s_1^2$	$t_5 = s_1 s_2$	
M_3	$t_3 = s_1$	$t_6 = s_2$	

and

The Optimal Schedule

Processors	Jobs		
M_1	$t_4 = s_1 s_2$	$t_2 = s_1^2$	$t_3 = s_1$
M_2	$t_5 = s_1 s_2$	$t_1 = s_2^2$	$t_6 = s_2$
M_3	$t_7 = s_1 + s_2 + 1.$		

Thus, if the processor M_2 is the latest finish processor, we get

$$T^{LS} = \frac{MT_2}{s_2} = (1+\alpha)(s_1 + s_2 + \frac{s_1 + s_2 + 1}{s_2}), \qquad T^* = s_1 + s_2 + 1,$$

and

$$\frac{T^{LS}}{T^*} = (1+\alpha)(1 + \frac{1}{s_2} - \frac{1}{s_1 + s_2 + 1}).$$

(iii) If the processor M_3 is the latest finish processor, then we let the set of processors is $M = \{M_3, M_1, M_2\}$.

a) As $s_1 \le s_2$, we set processing time and ready time of jobs are

Jobs J_i	J_1	J_2	J_3	J_4	J_5	J_6	J_7
Processing time t_i	s_1	s_1^2	s_2^2	s_2	$s_1 s_2$	$s_1 s_2$	$s_1 + s_2 + 1$

and

Ready Time

$w(0,1) = \alpha s_1$	$w(1,4) = \alpha s_2$
$w(0,2) = \alpha s_1^2$	$w(2,5) = \alpha s_1 s_2$
$w(0,3) = \alpha s_2^2$	$w(3,6) = \alpha s_1 s_2$
$w(4,7) = \alpha(s_1 + s_2 + 1)$	$w(i,j) = 0$ for others i,j.

Then in this case, the LS schedule and the optimal schedule are

The LS Schedule

Processors	Jobs		
M_3	$t_1 = s_1$	$t_4 = s_2$	$t_7 = s_1 + s_2 + 1$
M_1	$t_2 = s_1^2$	$t_5 = s_1 s_2$	
M_2	$t_3 = s_2^2$	$t_6 = s_1 s_2$	

and

The Optimal Schedule

Processors	Jobs		
M_1	$t_5 = s_1 s_2$	$t_2 = s_1^2$	$t_1 = s_1$
M_2	$t_6 = s_1 s_2$	$t_3 = s_2^2$	$t_4 = s_2$
M_3	$t_7 = s_1 + s_2 + 1.$		

b) As $s_1 > s_2$, we set processing time and ready time of jobs are

Jobs J_i	J_1	J_2	J_3	J_4	J_5	J_6	J_7
Processing time t_i	s_1	s_1^2	s_2^2	s_1s_2	s_2	s_1s_2	s_1+s_2+1

and

Ready Time

$w(0,1)=\alpha s_1$	$w(1,5)=\alpha s_2$
$w(0,2)=\alpha s_1^2$	$w(2,6)=\alpha s_1s_2$
$w(0,3)=\alpha s_2^2$	$w(3,4)=\alpha s_1s_2$
$w(5,7)=\alpha(s_1+s_2+1)$	$w(i,j)=0$ for others i,j.

Then in this case, the LS schedule and the optimal schedule are

The LS Schedule

Processors	Jobs		
M_3	$t_1=s_1$	$t_5=s_2$	$t_7=s_1+s_2+1$
M_1	$t_2=s_1^2$	$t_6=s_1s_2$	
M_2	$t_3=s_2^2$	$t_4=s_1s_2$	

and

The Optimal Schedule

Processors	Jobs		
M_1	$t_4=s_1s_2$	$t_2=s_1^2$	$t_1=s_1$
M_2	$t_6=s_1s_2$	$t_3=s_2^2$	$t_5=s_2$
M_3	$t_7=s_1+s_2+1.$		

Thus, if the processor M_3 is the latest finish processor, we get

$$T^{LS}=MT_3=(1+\alpha)(2s_1+2s_2+1),\qquad T^*=s_1+s_2+1,$$

and

$$\frac{T^{LS}}{T^*}=(1+\alpha)(2-\frac{1}{s_1+s_2+1}).$$

Thus, the above example shows that the bound given in Theorem 3.1 is tight for $m=3$.

(2) As $m>3$, we assume that $s_m=1$, $s_i\geq 1$, $i=1,2,\cdots,m-1$, and let the set of jobs is

Line	1	2	3	$\cdots$	$m-2$	$m-1$	m
Line 1	s_1^2	s_2^2	s_3^2	$\cdots$	s_{m-2}^2	s_{m-1}^2	s_{m-1}
Line 2	s_1s_2	s_2s_3	s_3s_4	$\cdots$	$s_{m-2}s_{m-1}$	$s_{m-1}s_1$	s_{m-2}
Line 3	s_1s_3	s_2s_4	s_3s_5	$\cdots$	$s_{m-2}s_1$	$s_{m-1}s_2$	s_{m-3}
Line 4	s_1s_4	s_2s_5	s_3s_6	$\cdots$	$s_{m-2}s_2$	$s_{m-1}s_3$	s_{m-4}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
Line i	s_1s_i	s_2s_{i+1}	s_3s_{i+2}	$\cdots$	$s_{m-2}s_{i-2}$	$s_{m-1}s_{i-1}$	s_{m-i}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
Line m-1	s_1s_{m-1}	s_2s_1	s_3s_2	$\cdots$	$s_{m-2}s_{m-3}$	$s_{m-1}s_{m-2}$	s_1
Line m	$\sum_{i=1}^{m-1} s_i+1.$						

It follows that the processing time of the lasted finish job is $\sum_{i=1}^{m-1} s_i+1$, the others are $s_i s_j$, $i=1,2,\cdots,m$, $j=1,2\cdots,m-1$, and the total number of jobs is $m(m-1)+1$.

In short, by adjusting the order between processors and jobs according to the latest finish processor and the value of s_i, we can get an example so that the last job is assigned to the latest finish processor M_k and the ready time $w(0,j)=\alpha t_j$, $j=1,2,\cdots,m$. If some processor processes t_j is after t_i, then we set $w(i,j)=\alpha t_j$ and $w(*,m(m-1)+1)=\alpha\sum_{i=1}^{m} s_i$. Otherwise, we set $w(i,j)=0$ so that each job needs the ready time in the LS schedule. However, each job does not need the ready time in the optimal schedule through adjusting the order of jobs.

Thus, the LS schedule of this example is

The LS Schedule

Processors	Jobs					
M_k	s_k^2	$s_k s_{k+1}$	$s_k s_{k+2}$	$\cdots$	$s_k s_{k-1}$	$\sum_{i=1}^{m-1} s_i+1$
M_1	s_1^2	$s_1 s_2$	$s_1 s_3$	$\cdots$	$s_1 s_{m-1}$	
M_2	s_2^2	$s_2 s_3$	$s_2 s_4$	$\cdots$	$s_2 s_1$	
M_3	s_3^2	$s_3 s_4$	$s_3 s_5$	$\cdots$	$s_3 s_2$	
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	
M_{m-1}	s_{m-1}^2	$s_{m-1} s_1$	$s_{m-1} s_2$	$\cdots$	$s_{m-1} s_{m-2}$	
M_m	s_{m-1}	s_{m-2}	s_{m-3}	$\cdots$	s_1	

and the optimal schedule of this example is

The Optimal Schedule

Processors	Jobs					
M_1	$s_1 s_2$	s_1^2	$s_1 s_3$	$\cdots$	$s_1 s_{m-1}$	s_1
M_2	$s_2 s_3$	s_2^2	$s_2 s_4$	$\cdots$	$s_2 s_1$	s_2
M_3	$s_3 s_4$	s_3^2	$s_3 s_5$	$\cdots$	$s_3 s_2$	s_3
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
M_k	$s_k s_{k+1}$	s_k^2	$s_k s_{k+2}$	$\cdots$	$s_k s_{k-1}$	s_k
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
M_{m-1}	$s_{m-1} s_1$	s_{m-1}^2	$s_{m-1} s_2$	$\cdots$	$s_{m-1} s_{m-2}$	s_{m-1}
M_m	$\sum_{i=1}^{m-1} s_i+1.$					

In this example, we have

$$T^{LS}=\frac{MT_k}{s_k}=(1+\alpha)\Big(\sum_{i=1}^{m-1} s_i+\frac{\sum_{i=1}^{m-1} s_i+1}{s_k}\Big),\qquad T^*=\sum_{i=1}^{m} s_i,$$

and

$$\frac{T^{LS}}{T^*}=\frac{MT_k}{s_k}=(1+\alpha)\Big(1+\frac{1}{s_k}-\frac{1}{\sum_{i=1}^{m} s_i}\Big).$$

Therefore, the above examples show that the bound given in Theorem 3.1 is tight for any $m\ge 3$. This completes the proof the theorem. □

Remark 3.2. Note that for the optimal solution T^*, we always have

$$T^* \geq \frac{t_j}{s_k},$$

where t_j is the processing time of the latest finish job J_j and s_k is the speed of the latest finish processor. Theorem 3.1 shows that if the optimal solution T^* is bigger than the processing time t_j of the latest finish job J_j, then the bound $(1+\alpha)(1+\frac{1}{s_k}-\frac{1}{\sum_{i=1}^{m} s_i})$ of the ratio of the approximate solution T^{LS} to the optimal solution T^* is tight for any $m \geq 3$ under the LS algorithm.

As special cases of Theorem 3.1, we have

Corollary 3.3. *The scheduling problem in Theorem 3.1 under the LS algorithm has the bound*

$$\frac{T^{LS}}{T^*} \leq (1+\alpha)(2-\frac{1}{\sum_{i=1}^{m} s_i}),$$

where s_k is the speed of the latest finish processor. Moreover, if $s_k = 1$, then this bound is tight for any $m \geq 3$.

Corollary 3.4. *If the ready time of every job is 0 in Theorem 3.1, i.e., all $w(*,*) = 0$, then the scheduling problem under the LS algorithm has the bound*

$$\frac{T^{LS}}{T^*} \leq 2-\frac{1}{\sum_{i=1}^{m} s_i}$$

where s_k is the speed of the latest finish processor. Moreover, if $s_k = 1$, then this bound is tight for any $m \geq 3$.

We now present an upper bound of the ratio of the approximate solution T^{LS} to the optimal solution T^* without making any assumptions.

Theorem 3.5. *Consider the problem of scheduling n jobs $\{J_1,\ J_2,\ \cdots,\ J_n\}$ on m uniform processors $\{M_1,\ M_2,\ \cdots,\ M_m\}$ provided each job has a ready time. Given the ready time $w(j,l)$ and processing time t_l of J_l for $j,\ l = 1,2,\cdots,n$, and let*

$$\alpha = \max_{j,\,l=1,2,\cdots,n}\{\frac{w(j,l)}{t_l}\}.$$

Then the bound of this scheduling problem under the LS algorithm is

$$\frac{T^{LS}}{T^*} \leq (1+\alpha)(2-\frac{s_k}{\sum_{i=1}^{m} s_i})$$

for any $m \geq 3$, where s_k is the speed of the latest finish processor.

Proof. Based on the *LS* algorithm introduced in Section 2, we may assume that some processor M_k $(1 \leq k \leq m)$ is the latest finish processor and the latest job is J_j $(1 \leq j \leq n)$. Then on the processor M_k, we have

$$T^{LS} = \frac{MT_k}{s_k}. \tag{3.9}$$

On other processors, we find

$$\frac{MT_i}{s_i} \geq \frac{MT_k - (w(*,j) + t_j)}{s_k}, i = 1, 2, \cdots, m, i \neq k. \tag{3.10}$$

Thus

$$\begin{aligned}\sum_{i=1}^{m} MT_i &= MT_k + \sum_{\substack{i=1\\ i\neq k}}^{m} MT_i \\ &\geq s_k T^{LS} + \sum_{\substack{i=1\\ i\neq k}}^{m} s_i T^{LS} - \frac{1}{s_k}(w(*,j) + t_j) \sum_{\substack{i=1\\ i\neq k}}^{m} s_i \\ &= \sum_{i=1}^{m} s_i T^{LS} - \frac{1}{s_k}(w(*,j) + t_j) \sum_{\substack{i=1\\ i\neq k}}^{m} s_i. \end{aligned} \tag{3.11}$$

On the other hand, for the optimal solution T^*, we have

$$T^* \geq \frac{t_j}{s_k} \tag{3.12}$$

and

$$T^* \geq \frac{\sum_{i=1}^{n} t_i}{\sum_{i=1}^{m} s_i}. \tag{3.13}$$

By the definition of α and (3.12), we get

$$w(*,j) + t_j \leq (1+\alpha) t_j \leq (1+\alpha) s_k T^*. \tag{3.14}$$

Then, by (3.11) and (3.14), we obtain

$$\sum_{i=1}^{m} MT_i \geq \sum_{i=1}^{m} s_i T^{LS} - (1+\alpha) T^* \sum_{\substack{i=1\\ i\neq k}}^{m} s_i. \tag{3.15}$$

In view of the definition of α and (3.13), we deduce

$$\begin{aligned}\sum_{i=1}^{m} MT_i &= \sum_{i=1}^{m} \sum_{\{t_h\}\in ML_i} (w(*,h) + t_h) \\ &= \sum_{h=1}^{n} (w(*,h) + t_h) \\ &\leq (1+\alpha) \sum_{h=1}^{n} t_h \\ &\leq (1+\alpha) T^* \sum_{i=1}^{m} s_i. \end{aligned} \tag{3.16}$$

Using (3.15) and (3.16), we have

$$(1+\alpha)T^*\sum_{i=1}^{m} s_i \geq \sum_{i=1}^{m} MT_i \geq T^{LS}\sum_{i=1}^{m} s_i - (1+\alpha)T^*\sum_{\substack{i=1\\ i\neq k}}^{m} s_i.$$

This implies

$$(1+\alpha)(\sum_{i=1}^{m} s_i + \sum_{\substack{i=1\\ i\neq k}}^{m} s_i)T^* \geq T^{LS}\sum_{i=1}^{m} s_i.$$

Therefore

$$\begin{aligned} \frac{T^{LS}}{T^*} &\leq \frac{(1+\alpha)}{\sum\limits_{i=1}^{m} s_i}(\sum_{i=1}^{m} s_i + \sum_{\substack{i=1\\ i\neq k}}^{m} s_i) \\ &= \frac{(1+\alpha)}{\sum\limits_{i=1}^{m} s_i}[\sum_{i=1}^{m} s_i + (\sum_{i=1}^{m} s_i - s_k)] \\ &= (1+\alpha)(2-\frac{s_k}{\sum\limits_{i=1}^{m} s_i}). \end{aligned}$$

This completes the proof of the theorem. □

Remark 3.6. Theorem 3.5 shows that the approximate solution T^{LS} is less than $(1+\alpha)(2-\frac{s_k}{\sum_{i=1}^{m} s_i})$ times of the optimal solution T^* under the LS algorithm without making any assumptions.

As special cases of Theorem 3.5, we have

Corollary 3.7. *The scheduling problem in Theorem 3.5 under the LS algorithm has the bound*

$$\frac{T^{LS}}{T^*} \leq (1+\alpha)(2-\frac{1}{\sum\limits_{i=1}^{m} s_i})$$

for any $m \geq 3$, where s_k is the speed of the latest finish processor.

Corollary 3.8. *If the ready time of every job is 0 in Theorem 3.5, i.e., all $w(*,*)=0$, then the scheduling problem under the LS algorithm has the bound*

$$\frac{T^{LS}}{T^*} \leq 2-\frac{s_k}{\sum\limits_{i=1}^{m} s_i}$$

for any $m \geq 3$, where s_k is the speed of the latest finish processor.

References

[1] W. Ding, Scheduling problem on general purpose machinery and two groups tasks with uniform processors, *Acta Sci. Natur. Univ. Sunyatseni* 43 (2004) 33–36.

[2] W. Ding, A scheduling problem on a general-purpose machine and three groups of tasks with identical processors, *J. Shanghai Univ. Nat. Sci.* 11 (2005) 48–51.

[3] W. Ding, Sequencing of four groups of workpieces on general-purpose machinery, *J. South China Univ. Tech. Nat. Sci.*, 33 (2005) 108–111.

[4] W. Ding, A type of scheduling problem on general-purpose machinery and n group tasks, *OR Transactions* 10 (2006) 122–126.

[5] W. Ding, A type of scheduling problem on m general-purpose machinery and n group tasks with uniform processors, *Acta Sci. Natur. Univ. Sunyatseni* 47 (2008) 19–22.

[6] M. Gairing, B. Monien and A. Woclaw, A faster combinatorial approximation algorithm for scheduling unrelated parallel machines, *Theoret. Comput. Sci.* 380 (2007) 87–99.

[7] R.L. Graham, Bounds on multiprocessing timing anomalies, *SIAM J. Appl. Math.* 17 (1969) 416–429.

[8] C. Imreh, Scheduling problems on two sets of identical machines, *Computing* 70 (2003) 277–294.

[9] I.M. Ovacik and R. Uzsoy, Worst-case error bounds for parallel machine sceduling problems with bounded sequence dependent setup times, *Oper. Res. Lett.* 14 (1993, 251–256.

[10] P. Schuurman and G.J. Woeginger, Polynomial time approximation algorithms for machine scheduling: Ten open problems, *J. Sched.* 2 (1999) 203–213.

Manuscript received 22 November 2009
revised 8 May 2010
accepted for publication 9 December 2010

Wei Ding
Department of Mathematics, Sun Yat-sen University,
510275 Guangzhou, China
E-mail address: dingwei@mail.sysu.edu.cn

Yi Zhao
Department of Mathematics, Sun Yat-sen University,
510275 Guangzhou, China
E-mail address: stszy@mail.sysu.edu.cn