PAFI

# The makespan problem of scheduling multi groups of jobs on multi processors at different speeds

## Wei Ding

Department of Mathematics, Sun Yat-sen University, 510275 Guangzhou, China

### Abstract

*In the paper we mainly study the makespan problem of scheduling $n$ groups of jobs on $n$ special-purpose processors and $m$ general-purpose processors at different speeds. We first propose an improved LPT algorithm and investigate several properties of this algorithm. We then obtain an upper bound for the ratio of the approximate solution $T$ to the optimal solution $T^*$ under the improved LPT algorithm.*

*Key words:* Mathematics Subject Classification (2000): 90B35, 68M20
Heuristic algorithm, LPT algorithm, approximate solutions, optimal solutions, upper bound.

## 1. Introduction

The problem of scheduling $n$ jobs $\{J_1, J_2, \cdots, J_n\}$ with given processing time on $m$ identical 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 identical processors at different speeds, 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 identical processors.

*Email:* Wei Ding [dingwei@mail.sysu.edu.cn].

Gairing et al. (2007, [6]) proposed a simple combinatorial algorithm for the problem of scheduling $n$ jobs on $m$ processors at different speeds to minimize a cost stream and showed it is effective and of low complexity.

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 problem of scheduling multi groups of jobs on multi processors at same or different speeds were studied provided each job has no ready time. In 2006 Ding [1] studied the problem of scheduling $n$ groups of jobs on one special-purpose processor and $n$ general-purpose processors at same speeds under an improved LPT algorithm. In 2008 Ding [2] investigated the problem of scheduling $n$ groups of jobs on $n$ special-purpose processors and $m$ general-purpose processors at same speeds under an improved LPT algorithm. In 2009 Ding [3] present an improved LS algorithm for the $Q_{m+2}/r_j/C_{max}$ scheduling problem on $m$ general-purpose processors and two special-purpose processors. In 2010 Ding [4] studied a heuristic algorithm of the $Q//C_{max}$ problem on multi-tasks with uniform processors. More recently, Ding and Zhao [5] investigated an improved LS algorithm for the problem of scheduling multi groups of jobs on multi processors at the same speed provided each job has a ready time.

However, the problem of scheduling $n$ groups of jobs on $n$ special-purpose processors and $m$ general-purpose processors at different speeds has not been studied yet.

Note that the classical LPT algorithm is only useful to solve the problem of scheduling one group of jobs on multi processors at same speeds or different speeds. Therefore, our aim of this study is to propose an improved LPT algorithm based on the classical LPT algorithm and to use this new algorithm to analyze this problem provided processors have different speeds.

The remainder of the paper is organized as follows. In Section 2, we proposed an improved LPT algorithm and study several properties of the improved LPT algorithm. In Section 3 we obtain an upper bound for the ratio of the approximate solution $T$ to the optimal solution $T^*$ under the improved $LPT$ algorithm.

## 2. An improved $LPT$ algorithm

In the section, we will propose an improved LPT algorithm for this scheduling problem and then investigate several properties of this algorithm.

We will use the following notations throughout the remainder of the paper.

Let $L_i$ $(i = 1, \cdots, n)$ denote the $i$th group of jobs, and let $M_i$ $(i = 1, \cdots, n)$ and $M_{n+j}$ $(j = 1, \cdots, m)$ denote the $i$th special-purpose processor and the $j$th general-purpose processor, respectively. Then, let $L = (L_1, L_2, \cdots, L_n)$ stand for the set of all groups of jobs and let $|L_r|$ denote the number of all jobs in $L_r$. Finally, let $|L| = |L_1| + |L_2| + \cdots + |L_n|$ denote the number of all jobs of all groups.

Let $J_{rk}$ denote the $k^{th}$ job in the $r^{th}$ group after ordering. If the job $J_{rk}$ is earlier than $J_{r'k'}$ to be assigned to a processor, then we write $J_{rk} \prec J_{r'k'}$. If the job $J_{rk}$ is assigned to the processor $M_l$, then we write $J_{rk} \in M_l$.

We use $t_{ri}$ $(r = 1, \cdots, n; i = 1, \cdots, |L_r|)$ to denote the processing time of $J_{ri}$. Then, we denote by $s_i$ $(i = 1, \cdots, n)$ the speed of the special–purpose processor $M_i$ and by $s_{n+j}$ $(j = 1, \cdots, m)$ the speed of the general–purpose processor $M_{n+j}$, respectively.

Note that the speeds of general–purpose processors are less than those of special–purpose processors in real production systems. For simplicity, we take $s_{n+j} = 1$ $(1 \le j \le m)$ and assume $s_i \ge 1$ $(i = 1, \cdots, n)$.

Let $MT_l(J_{rk})$ denote the latest absolutely finish time of the processor $M_l$ before the job $J_{rk}$ is assigned and let $MT_l$ denote the latest absolutely finish time of the processor $M_l$ after all jobs are assigned. Next, let $ML_l(J_{rk})$ $(l = 1, 2, \cdots, m + n)$ denote the set of jobs assigned in the processor $M_l$ before the job $J_{rk}$ is assigned and let

$$T_r = \sum_{i=1}^{|L_r|} t_{ri} \quad r = 1, 2, \cdots, n,$$

$$MT_l(J_{rk}) = \sum_{J_{r'k'} \in M_l, J_{r'k'} \prec J_{rk}} t_{r'k'}, \quad MT_l$$

$$= \sum_{J_{rk} \in M_l} t_{rk}, \quad l = 1, 2, \cdots, n + m,$$

and

$$ML_l(J_{rk}) = \{J_{r'k'} | J_{r'k'} \prec J_{rk}, J_{r'k'} \in M_l\}$$
$$l = 1, 2, \cdots, n + m.$$

The main strategy of the improved $LPT$ algorithm is based on the intuitive fact that $n$ groups are listed in order of the total real processing time of the group, i.e., $\frac{T_1}{s_1} \ge \frac{T_2}{s_2} \ge \cdots \ge \frac{T_n}{s_n}$, that the jobs in each group are listed in order of the total processing time of the job, i.e., $t_{ri} \ge t_{ri+1}$, $r = 1, 2, \cdots, n$, $i = 1, 2, \cdot, |L_r| - 1$, and that whenever a processor becomes idle for assignment, the first job unexecuted is taken from the list and assigned to this processor.

Assume that the job is assigned in an increasing order of the index and that if all jobs before the $k_r^{th}$ job in the group $L_r$ have been assigned and the job $J_{rk_r}$ is waiting for being assigned, then jobs $J_{1k_1}, J_{2k_2}, \cdots, J_{nk_n}$ are called as candidates.

**Definition 1.** *When jobs $J_{1k_1}, J_{2k_2}, \cdots, J_{nk_n}$ are candidates, the possible absolutely processing time (SMT) of the special-purpose processor for the group $L_r$ is*

$$SMT_r(k_r) = T_r - \sum_{\substack{J_{ri} \in \bigcup_{j=n+1}^{n+m} M_j, i < k_r}} t_{ri},$$
$$r = 1, 2, \cdots, n.$$

*When some group $L_r$ is the empty set, we set $SMT_r = SMT_r(k) \equiv 0$, where $k$ is an arbitrary positive integer.*

**Definition 2.** *When the job $J_{rk_r}$ is the candidate, the relative SMT of the group $L_r$ $(r = 1, 2, \cdots, n)$ is*

$$\frac{SMT_r(k_r)}{s_r}, \quad r = 1, 2, \cdots, n.$$

The steps of the improved $LPT$ algorithm are the following:

Step 1. Ordering. Let $T_1/s_1 \ge T_2/s_2 \ge \cdots \ge T_n/s_n$, $t_{ri} \ge t_{ri+1}$, $i = 1, 2, \cdots, |L_r| - 1$, $r = 1, 2, \cdots, n$.

Step 2. Initialization. Set $k_r = 1$, $MT_l(J_{rk_r}) = 0$, and $ML_l(J_{rk_r}) = \emptyset$, $r = 1, 2, \cdots, n$, $l = 1, 2, \cdots, n+m$.

Step 3. Choose the job for processing according to the rule of the maximum relative SMT. If

$$r = \min\{r' \,|\, \frac{SMT_{r'}(k_{r'})}{s_{r'}} = \max_{r''=1,2,\cdots,n} \frac{SMT_{r''}(k_{r''})}{s_{r''}}\}, \tag{1}$$

then the job $J_{rk_r}$ is the candidate.

Step 4. Choose the processor according to the rule of being the first with the earlier idle time. When the job $J_{rk_r} \in L_r$ ($r = 1, 2, \cdots, n$) is waiting for being assigned, if

$$p = \min\{q \,|\, \frac{MT_q(J_{rk_r}) + t_{rk_r}}{s_q}$$

$$= \min_{l=r,n+1,\cdots,n+m} \frac{MT_l(J_{rk_r}) + t_{rk_r}}{s_l}\},$$

then let $J_{rk_r} \in M_p$.

Step 5. If all jobs are assigned, then the program is over. Otherwise, go to Step 3.

Let $ST(J_{ij})$ and $CT(J_{ij})$ denote the beginning time and the finishing time of the job $J_{ij}$, respectively. We now present several properties of the improved $LPT$ algorithm.

**Lemma 1.** *(1) If $J_{ij}, J_{rk} \in M_l$, $l = 1, 2, \cdots, n+m$, and $J_{ij} \prec J_{rk}$, then*

$$CT(J_{ij}) \le ST(J_{rk}).$$

*(2) If $J_{ij}, J_{rk} \in L_r$, $r = 1, 2, \cdots, n$, and $J_{ij} \prec J_{rk}$, then $CT(J_{ij}) - t_{ij} \le ST(J_{rk})$.*
*(3) If $J_{ij} \in L_r$, $r = 1, 2, \cdots, n$, then $\frac{MT_l(J_{ij}) + t_{ij}}{s_l} \ge CT(J_{ij})$, $l = r, n+1, \cdots, n+m$.*

*Proof.* By Step 1 and the definitions of $ST(J_{ij})$ and $CT(J_{ij})$, we get (1) and (2). By Step 4 and the definitions of $ST(J_{ij})$ and $CT(J_{ij})$, we obtain (3). This completes the proof of the lemma. □

**Lemma 2.** *Let $T$ be the makespan of the above improved $LPT$ algorithm. If there exists a job $J_{rp} \in L_r$ such that $CT(J_{rp}) = T$, $r = 1, 2, \cdots, n$, $p = 1, 2, \cdots, |L_r|$ and $J_{kq} \prec J_{rp}$, $k = 1, 2, \cdots, n$, $k \ne r$, $q = 1, 2, \cdots, |L_k|$, then*

$$T_k \ge SMT_k(q) > s_k T.$$

*Proof.* Because $J_{kq} \prec J_{rp}$, we may assume the job $J_{kq}$ is chosen to assign when $J_{kq}$ and $J_{rs}$ are candidates, where $s \le p$. Based on the algorithm, we obtain

$$\frac{SMT_k(q)}{s_k} > \frac{SMT_r(s)}{s_r}.$$

By the definition of $SMT$, we have

$$\frac{T_k}{s_k} > \frac{SMT_k(q)}{s_k} > \frac{SMT_r(s)}{s_r} \ge \frac{SMT_r(p)}{s_r}.$$

If $J_{rp} \in M_r$, in view of $CT(J_{rp}) = T$ and $T$ being the makespan, then $MT_r = s_r T$. From the definition of $SMT_r(p)$, we know that $J_{rp}$ is the last finish job, but may not be the last assigned job of the group $L_r$. When the job $J_{rp}$ is waiting for being assigned, we have

$$SMT_r(p) = T_r - \sum_{J_{ri} \in \bigcup_{j=n+1}^{n+m} M_j,\, i<p} t_{rj}$$

$$= \sum_{J_{ri} \in M_r,\, i<p} t_{rj} + t_{rp} + t_{rp+1} + \cdots + t_{r|L_r|},$$

and

$$MT_r = \sum_{J_{ri} \in M_r,\, i<p} t_{rj} + t_{rp}.$$

Then it follows that

$$SMT_r(p) \ge MT_r = s_r T.$$

Thus, we get

$$T_k \ge SMT_k(q) > s_k T.$$

If $J_{rp} \in M_j$ ($n+1 \le j \le n+m$), in view of Step 4, then we have

$$\frac{MT_r(J_{rp}) + t_{rp}}{s_r} > MT_j(J_{rp}) + t_{rp} = T. \tag{2}$$

It follows that $MT_r(J_{rp}) + t_{rp} > s_r T$.

Note that $CT(J_{rp}) = T$. Thus

$$SMT_r(p) = MT_r(J_{rp}) + \sum_{i \ge p} t_{ri}$$

$$= MT_r(J_{rp}) + t_{rp} + \sum_{i \ge p+1} t_{ri}$$

$$\ge MT_r(J_{rp}) + t_{rp} > s_r T. \tag{3}$$

Therefore, we get $T_k > SMT_k(q) > s_k T$. This completes the proof of the lemma. □

**Lemma 3.** *If there exists a job $J_{rp} \in L_r$ in $L = (L_1, L_2, \cdots, L_n)$ such that $CT(J_{rp}) = T(L)$ and there exists at least one group $L_k$, $k \ne r$, such that*

$$\{J_{kq} | J_{kq} \in \bigcup_{j=n+1}^{n+m} M_j, J_{kq} \prec J_{rp}\} = \emptyset,$$

*then there exists some $L'$ such that $|L'| < |L|$ and*

$$T(L')/T^*(L') \geq T(L)/T^*(L) = T/T^*,$$

*where $T(L) = T$ and $T^*(L) = T^*$.*

*Proof.* Note that the assumption
$\{J_{kq}|J_{kq} \in \bigcup_{j=n+1}^{n+m} M_j, J_{kq} \prec J_{rp}\} = \emptyset$ means that all
assigned jobs in $L_k$ before the last finish job $J_{rp}$ have
not been assigned on the general-purpose processor $M_j$
$(n+1 \leq j \leq n+m)$. Let

$$L'_1 = L_1, \ L'_2 = L_2, \cdots, L'_{k-1} = L_{k-1}, \ L'_k = L_{k+1},$$

$$\cdots, L'_{n-1} = L_n, \ L'_n = \emptyset, \ L' = (L'_1, L'_2, \cdots, L'_n).$$

Then $SMT_n(L') \equiv 0$ and the order of the jobs in $L'_1$,
$L'_2, \cdots, L'_{n-1}$ is the same as that in $L - L_k$. Thus, they
have the same assignment.

By the assumption of Lemma 3, we know that all as-
signed jobs in $L_k$ before the job $J_{rp}$ have been assigned
on the special-purpose processor $M_k$. This implies that
any assigned jobs in $L_k$ after the last finish job $J_{rp}$ will
not change the last finish time $T(L)$. Scheduling $L - L_k$
on $n + m$ processors is equivalent to scheduling $L$ on
$n + m$ processors. Therefore the last finish time of $L$ is
the same as that of $L'$, i.e.

$$CT(L'|J_{rp}) = CT(L|J_{rp}) = T(L).$$

Since $|L'| < |L|$, it follows that

$$T^*(L') \leq T^*(L).$$

This yields

$$T(L')/T^*(L') \geq T(L)/T^*(L).$$

This completes the proof of the lemma.      $\square$

## 3. Analysis of the improved $LPT$ algorithm

In the section, we obtain an upper bound for the ratio
of the approximate solution $T$ to the optimal solution
$T^*$ under the improved $LPT$ algorithm.

**Theorem 1.** *Consider the problem of schedul-
ing $n$ groups of jobs $L = \{L_1, L_2, \cdots, L_n\}$ on
$\{M_1, M_2, \cdots, M_n\}$ special-purpose processors and
$\{M_{n+1}, M_{n+2} \cdots, M_{n+m}\}$ general-purpose proces-
sors at different speeds with the objective of minimizing*

*the makespan. Let $T$ be the makespan of the above im-
proved LPT algorithm. Then the bound of this schedul-
ing problem under the improved LPT algorithm is*

$$\frac{T}{T^*} \leq 1 + \frac{m}{\sum\limits_{i \in I} s_i},$$

*where $I$ is the set of group of jobs in which there exists
at least one job to be assigned on some general-purpose
processor before the latest finish time.*

*Proof.* Assume there exists $J_{rp} \in L_r$ such that
$CT(J_{rp}) = T$.
Case A. If $|I| = n$, i.e., $\forall J_{kq} \in L_k, k \neq r$,
$\{J_{kq}|J_{kq} \in \bigcup_{j=n+1}^{n+m} M_j, J_{kq} \prec J_{rp}\} \neq \emptyset$, then we may
assume

$$u_k = \max\{i|J_{ki} \in \bigcup_{j=n+1}^{n+m} M_j, J_{ki} \prec J_{rp}\}.$$

From the algorithm $J_{k1} \in M_k$, we know that $u_k \geq 2$.
Note that $CT(J_{rp}) = T$. By Lemma 1, we obtain

$$\frac{MT_r(J_{rp}) + t_{rp}}{s_r} \geq CT(J_{rp}) = T.$$

By Lemma 2, for any $J_{ku_k} \in L_k, k \neq r$, we have

$$T_k \geq SMT_k(u_k) > s_k T.$$

Thus

$$T^* \geq \frac{T_1 + T_2 + \cdots + T_n}{m + \sum\limits_{i=1}^{n} s_i}$$

$$\geq \frac{T_1 + T_2 + \cdots + MT_r(J_{rp}) + t_{rp} + \cdots + T_n}{m + \sum\limits_{i=1}^{n} s_i}$$

$$\geq \frac{s_1 T + s_2 T + \cdots + s_r T + \cdots + s_n T}{m + \sum\limits_{i=1}^{n} s_i}$$

$$\geq \frac{\sum\limits_{i=1}^{n} s_i}{m + \sum\limits_{i=1}^{n} s_i} T.$$

This yields

$$\frac{T}{T^*} \leq 1 + \frac{m}{\sum\limits_{i=1}^{n} s_i}.$$

Case B. If $|I| < n$, i.e., there exists $J_{kq} \in L_k$, $k \neq r$, such that

$$\{J_{kq}|J_{kq} \in \bigcup_{j=n+1}^{n+m} M_j, J_{kq} \prec J_{rp}\} = \emptyset,$$

then by Lemma 3 and the definition of $I$, we know that there exists $L'$ and $|L'| = |I|$ such that

$$\{J_{kq}|J_{kq} \in \bigcup_{j=n+1}^{n+m} M_j, J_{kq} \prec J_{rp}\} \neq \emptyset,$$

$$\forall J_{kq} \in L'_k, \ k \neq r.$$

By Lemma 3, in view of the proof of case A, we have

$$\frac{T}{T^*} = \frac{T(L)}{T^*(L)} \leq \frac{T(L')}{T^*(L')} \leq 1 + \frac{m}{\sum_{i \in I} s_i}.$$

This completes the proof of the theorem.             □

As a consequence of Theorem 1, we have
**Corollary 1.** *The scheduling problem in Theorem 1 under the improved LPT algorithm has the bound $\frac{T}{T^*} \leq 1 + \frac{m}{|I|}$.*

Next, the following example will show how the improved $LPT$ algorithm works.

Consider the following scheduling problems.

Assume that there are three groups of jobs and each group separately owns one special-purpose processor and jointly owns two general-purpose processors.

Step 1. Ordering.

Let the jobs of the group $L_1$ be denoted by $J_{11}, J_{12}, J_{13}, J_{14}, J_{15}, J_{16}$, and let their absolutely processing time be $t_{11} = 65, t_{12} = 42, t_{13} = 37, t_{14} = 36, t_{15} = 28, t_{16} = 22$, respectively.

Let the jobs of the group $L_2$ be denoted by $J_{21}, J_{22}, J_{23}, J_{24}, J_{25}$, and let their absolutely processing time be $t_{21} = 70, t_{22} = 55, t_{23} = 45, t_{24} = 39, t_{25} = 31$, respectively.

Let the jobs of the group $L_3$ be denoted by $J_{31}, J_{32}, J_{33}, J_{34}, J_{35}, J_{36}$, and let their absolutely processing time be $t_{31} = 60, t_{32} = 50, t_{33} = 40, t_{34} = 36, t_{35} = 34, t_{36} = 30$, respectively.

Let the speed of the special-purpose $M_1$ of $L_1$ be $s_1 = 1.2$, let the speed of the special-purpose $M_2$ of $L_2$ be $s_2 = 1.3$, and let the speed of the special-purpose $M_3$ of $L_3$ be $s_3 = 1.5$, respectively.

Let the speeds of two general-purpose processors be $s_4 = s_5 = 1$. Then $T_1 = 230, T_2 = 240, T_3 = 250, \frac{T_1}{s_1} = 191.7, \frac{T_2}{s_2} = 184.6, \frac{T_3}{s_3} = 166.7$.

Step 2. Initialization.

Set $k_1 = 1$, $k_2 = 1$, $k_3 = 1$. Let the latest absolutely finish time of all processors be $MT_l = 0$, and let the sets of jobs assigned in all processors be $ML_l = \emptyset$, $l = 1, 2, 3, 4, 5$.

Step 3. Choose the job for processing according to the rule of the maximum realtive $SMT$.

Since

$$\frac{SMT_1(1)}{s_1} = \frac{230}{1.2} = 191.7,$$

$$\frac{SMT_2(1)}{s_2} = \frac{240}{1.3} = 184.6,$$

and

$$\frac{SMT_3(1)}{s_3} = \frac{250}{1.5} = 166.7,$$

it follows that the job $J_{11}$ is the candidate. Take $k_1 = 2$.

Step 4. Choose the processor according to the rule of being the first with the earlier idle time.

Since

$$\frac{(MT_1 + t_{11})}{s_1} = \frac{65}{1.2} = 54.2,$$

$$\frac{(MT_4 + t_{11})}{s_4} = \frac{65}{1} = 65,$$

and

$$\frac{(MT_5 + t_{11})}{s_5} = \frac{65}{1} = 65,$$

it follows that the job $J_{11}$ is assigned on the processor $M_1$. Thus

$$ML_1 = \{J_{11}\}, \quad MT_1 = 65.$$

Step 3. Choose the job for processing.

Since

$$\frac{SMT_1(2)}{s_1} = \frac{230}{1.2} = 191.7,$$

$$\frac{SMT_2(1)}{s_2} = \frac{240}{1.3} = 184.6,$$

and

$$\frac{SMT_3(1)}{s_3} = \frac{250}{1.5} = 166.7,$$

it follows that the job $J_{12}$ is the candidate. Take $k_1 = 3$.

Step 4. Choose the processor.

Since

$$\frac{(MT_1 + t_{12})}{s_1} = \frac{(65 + 42)}{1.2} = 89.2,$$

$$\frac{(MT_4 + t_{12})}{s_4} = \frac{42}{1} = 42,$$

and
$$\frac{(MT_5 + t_{12})}{s_5} = \frac{42}{1} = 42,$$

it follows that the job $J_{12}$ is assigned on the processor $M_4$. Thus

$$ML_4 = \{J_{12}\}, \quad MT_4 = 42.$$

Step 3. Choose the job for processing.
Since
$$\frac{SMT_1(3)}{s_1} = \frac{(230 - 42)}{1.2} = \frac{188}{1.2} = 156.7,$$

$$\frac{SMT_2(1)}{s_2} = \frac{240}{1.3} = 184.6,$$

and
$$\frac{SMT_3(1)}{s_3} = \frac{250}{1.5} = 166.7,$$

it follows that the job $J_{21}$ is the candidate. Take $k_2 = 2$.
Step 4. Choose the processor.
Since
$$\frac{(MT_2 + t_{21})}{s_2} = \frac{70}{1.3} = 53.8,$$

$$\frac{(MT_4 + t_{21})}{s_4} = \frac{(42 + 70)}{1} = 112,$$

and
$$\frac{(MT_5 + t_{21})}{s_5} = \frac{70}{1} = 70,$$

it follows that the job $J_{21}$ is assigned on the processor $M_2$. Thus,

$$ML_2 = \{J_{21}\}, \quad MT_2 = 70.$$

Step 3. Choose the job for processing.
Since
$$\frac{SMT_1(3)}{s_1} = \frac{(230 - 42)}{1.2} = \frac{188}{1.2} = 156.7,$$

$$\frac{SMT_2(2)}{s_2} = \frac{240}{1.3} = 184.6,$$

and
$$\frac{SMT_3(1)}{s_3} = \frac{250}{1.5} = 166.7,$$

it follows that the job $J_{22}$ is the candidate. Take $k_2 = 3$.
Step 4. Choose the processor.
Since
$$\frac{(MT_2 + t_{22})}{s_2} = \frac{125}{1.3} = 96.2,$$

$$\frac{(MT_4 + t_{22})}{s_4} = \frac{97}{1} = 97,$$

and
$$\frac{(MT_5 + t_{22})}{s_5} = \frac{55}{1} = 55,$$

it follows that the job $J_{22}$ is assigned on the processor $M_5$. Thus

$$ML_5 = \{J_{22}\}, \quad MT_5 = 55.$$

Step 3. Choose the job for processing.
Since
$$\frac{SMT_1(3)}{s_1} = \frac{(230 - 42)}{1.2} = \frac{188}{1.2} = 156.7,$$

$$\frac{SMT_2(3)}{s_2} = \frac{(240 - 55)}{1.3} = 142.3,$$

and
$$\frac{SMT_3(1)}{s_3} = \frac{250}{1.5} = 166.7,$$

it follows that the job $J_{31}$ is the candidate. Take $k_3 = 2$.
Step 4. Choose the processor.
Since
$$\frac{(MT_3 + t_{31})}{s_3} = \frac{60}{1.5} = 40,$$

$$\frac{(MT_4 + t_{31})}{s_4} = \frac{(42 + 60)}{1} = 102,$$

and
$$\frac{(MT_5 + t_{31})}{s_5} = \frac{(55 + 60)}{1} = 115,$$

it follows that the job $J_{31}$ is assigned on the processor $M_3$. Thus

$$ML_3 = \{J_{31}\}, \quad MT_3 = 60.$$

Step 3. Choose the job for processing.
Since
$$\frac{SMT_1(3)}{s_1} = \frac{(230 - 42)}{1.2} = \frac{188}{1.2} = 156.7,$$

$$\frac{SMT_2(3)}{s_2} = \frac{(240 - 55)}{1.3} = 142.3,$$

and
$$\frac{SMT_3(2)}{s_3} = \frac{250}{1.5} = 166.7,$$

it follows that the job $J_{32}$ is the candidate. Take $k_3 = 3$.
Step 4. Choose the processor.

Since

$$\frac{(MT_3 + t_{32})}{s_3} = \frac{(60 + 50)}{1.5} = 73.3,$$

$$\frac{(MT_4 + t_{32})}{s_4} = \frac{(42 + 50)}{1} = 92,$$

and

$$\frac{(MT_5 + t_{32})}{s_5} = \frac{(55 + 50)}{1} = 105,$$

it follows that $J_{32}$ is assigned on the processor $M_3$. Thus,

$$ML_3 = \{J_{31}, J_{32}\}, \quad MT_3 = 60 + 50 = 110.$$

Step 3. Choose the job for processing.
Since

$$\frac{SMT_1(3)}{s_1} = \frac{(230 - 42)}{1.2} = \frac{188}{1.2} = 156.7,$$

$$\frac{SMT_2(3)}{s_2} = \frac{(240 - 55)}{1.3} = 142.3,$$

and

$$\frac{SMT_3(3)}{s_3} = \frac{250}{1.5} = 166.7,$$

it follows that the job $J_{33}$ is the candidate. Take $k_3 = 4$.
Step 4. Choose the processor.
Since

$$\frac{(MT_3 + t_{33})}{s_3} = \frac{(110 + 40)}{1.5} = 100,$$

$$\frac{(MT_4 + t_{32})}{s_4} = \frac{(42 + 40)}{1} = 82,$$

and

$$\frac{(MT_5 + t_{32})}{s_5} = \frac{(55 + 40)}{1} = 95,$$

it follows that the job $J_{33}$ is assigned on the processor $M_4$. Thus,

$$ML_4 = \{J_{12}, J_{33}\}, \quad MT_4 = 42 + 40 = 82.$$

Step 3. Choose the job for processing.
Since

$$\frac{SMT_1(3)}{s_1} = \frac{(230 - 42)}{1.2} = \frac{188}{1.2} = 156.7,$$

$$\frac{SMT_2(3)}{s_2} = \frac{(240 - 55)}{1.3} = 142.3,$$

and

$$\frac{SMT_3(4)}{s_3} = \frac{(250 - 40)}{1.5} = 140,$$

it follows that the job $J_{13}$ is the candidate. Take $k_1 = 4$.
Step 4. Choose the processor.
Since

$$\frac{(MT_1 + t_{13})}{s_1} = \frac{(65 + 37)}{1.2} = 85,$$

$$\frac{(MT_4 + t_{13})}{s_4} = \frac{(82 + 37)}{1} = 119,$$

and

$$\frac{(MT_5 + t_{13})}{s_5} = \frac{(55 + 37)}{1} = 92,$$

it follows that the job $J_{13}$ is assigned on the processor $M_1$. Thus,

$$ML_1 = \{J_{11}, J_{13}\}, \quad MT_1 = 65 + 37 = 102.$$

Step 3. Choose the job for processing.
Since

$$\frac{SMT_1(4)}{s_1} = \frac{(230 - 42)}{1.2} = \frac{188}{1.2} = 156.7,$$

$$\frac{SMT_2(3)}{s_2} = \frac{(240 - 55)}{1.3} = 142.3,$$

and

$$\frac{SMT_3(4)}{s_3} = \frac{(250 - 40)}{1.5} = 140,$$

it follows that the job $J_{14}$ is the candidate. Take $k_1 = 5$.
Step 4. Choose the processor.
Since

$$\frac{(MT_1 + t_{14})}{s_1} = \frac{(102 + 36)}{1.2} = 115,$$

$$\frac{(MT_4 + t_{14})}{s_4} = \frac{(82 + 36)}{1} = 118,$$

and

$$\frac{(MT_5 + t_{14})}{s_5} = \frac{(55 + 36)}{1} = 91,$$

it follows that the job $J_{14}$ is assigned on the processor $M_5$. Thus,

$$ML_5 = \{J_{22}, J_{14}\}, \quad MT_5 = 55 + 36 = 91.$$

Step 3. Choose the job for processing.
Since

$$\frac{SMT_1(5)}{s_1} = \frac{(230 - 42 - 36)}{1.2} = \frac{152}{1.2} = 126.7,$$

$$\frac{SMT_2(3)}{s_2} = \frac{(240 - 55)}{1.3} = 142.3,$$

and

$$\frac{SMT_3(4)}{s_3} = \frac{(250 - 40)}{1.5} = 140,$$

it follows that the job $J_{23}$ is the candidate. Take $k_2 = 4$.

Step 4. Choose the processor.
Since

$$\frac{(MT_2 + t_{23})}{s_2} = \frac{(70 + 45)}{1.3} = 88.5,$$

$$\frac{(MT_4 + t_{23})}{s_4} = \frac{(82 + 45)}{1} = 127,$$

and

$$\frac{(MT_5 + t_{23})}{s_5} = \frac{(91 + 45)}{1} = 136,$$

it follows that the job $J_{23}$ is assigned on the processor $M_2$. Thus

$$ML_2 = \{J_{21}, J_{23}\}, \quad MT_2 = 70 + 45 = 115.$$

Step 3. Choose the job for processing.
Since

$$\frac{SMT_1(5)}{s_1} = \frac{(230 - 42 - 36)}{1.2} = \frac{152}{1.2} = 126.7,$$

$$\frac{SMT_2(4)}{s_2} = \frac{(240 - 55)}{1.3} = 142.3,$$

and

$$\frac{SMT_3(4)}{s_3} = \frac{(250 - 40)}{1.5} = 140,$$

it follows that the job $J_{24}$ is the candidate. Take $k_2 = 5$.

Step 4. Choose the processor.
Since

$$\frac{(MT_2 + t_{24})}{s_2} = \frac{(115 + 39)}{1.3} = 118.5,$$

$$\frac{(MT_4 + t_{24})}{s_4} = \frac{(82 + 39)}{1} = 121,$$

and

$$\frac{(MT_5 + t_{24})}{s_5} = \frac{(91 + 39)}{1} = 130,$$

it follows that the job $J_{24}$ is assigned on the processor $M_2$. Thus,

$$ML_2 = \{J_{21}, J_{23}, J_{24}\}, \quad MT_2 = 115 + 39 = 154.$$

Step 3. Choose the job for processing.

Since

$$\frac{SMT_1(5)}{s_1} = \frac{(230 - 42 - 36)}{1.2} = \frac{152}{1.2} = 126.7,$$

$$\frac{SMT_2(5)}{s_2} = \frac{(240 - 55)}{1.3} = 142.3,$$

and

$$\frac{SMT_3(4)}{s_3} = \frac{(250 - 40)}{1.5} = 140,$$

it follows that $J_{25}$ is the candidate. Take $k_2 = 6$.

Step 4. Choose the processor.
Since

$$\frac{(MT_2 + t_{25})}{s_2} = \frac{(154 + 31)}{1.3} = 142.3,$$

$$\frac{(MT_4 + t_{25})}{s_4} = \frac{(82 + 31)}{1} = 113,$$

and

$$\frac{(MT_5 + t_{25})}{s_5} = \frac{(91 + 31)}{1} = 122,$$

it follows that the job $J_{25}$ is assigned on the processor $M_4$. Thus

$$ML_4 = \{J_{12}, J_{33}, J_{25}\}, \quad MT_4 = 82 + 31 = 113.$$

Step 3. Choose the job for processing.
Note that all jobs in $L_2$ have been assigned. By comparing

$$\frac{SMT_1(5)}{s_1} = \frac{(230 - 42 - 36)}{1.2} = \frac{152}{1.2} = 126.7$$

with

$$\frac{SMT_3(4)}{s_3} = \frac{(250 - 40)}{1.5} = 140,$$

we see that the job $J_{34}$ is the candidate. Take $k_3 = 5$.

Step 4. Choose the processor.
Since

$$\frac{(MT_3 + t_{34})}{s_3} = \frac{(110 + 36)}{1.5} = 97.3,$$

$$\frac{(MT_4 + t_{34})}{s_4} = \frac{(113 + 36)}{1} = 149,$$

and

$$\frac{(MT_5 + t_{34})}{s_5} = \frac{(91 + 36)}{1} = 127,$$

it follows that the job $J_{34}$ is assigned on the processor $M_3$. Thus,

$$ML_3 = \{J_{31}, J_{32}, J_{34}\}, \quad MT_3 = 110 + 36 = 146.$$

Step 3. Choose the job for processing.
Since

$$\frac{SMT_1(5)}{s_1} = \frac{(230 - 42 - 36)}{1.2} = \frac{152}{1.2} = 126.7$$

and

$$\frac{SMT_3(5)}{s_3} = \frac{(250 - 40)}{1.5} = 140,$$

it follows that the job $J_{35}$ is the candidate. Take $k_3 = 6$.
Step 4. Choose the processor.
Since

$$\frac{(MT_3 + t_{35})}{s_3} = \frac{(146 + 34)}{1.5} = 120,$$

$$\frac{(MT_4 + t_{35})}{s_4} = \frac{(113 + 34)}{1} = 147,$$

and

$$\frac{(MT_5 + t_{35})}{s_5} = \frac{(91 + 34)}{1} = 125,$$

it follows that the job $J_{35}$ is assigned on the processor $M_3$. Thus,

$$ML_3 = \{J_{31}, J_{32}, J_{34}, J_{35}\}, \quad MT_3 = 146 + 34 = 180.$$

Step 3. Choose the job for processing.
Since

$$\frac{SMT_1(5)}{s_1} = \frac{(230 - 42 - 36)}{1.2} = \frac{152}{1.2} = 126.7$$

and

$$\frac{SMT_3(6)}{s_3} = \frac{(250 - 40)}{1.5} = 140,$$

it follows that the job $J_{36}$ is the candidate. Take $k_3 = 7$.
Step 4. Choose the processor.
Since

$$\frac{(MT_3 + t_{36})}{s_3} = \frac{(180 + 30)}{1.5} = 140,$$

$$\frac{(MT_4 + t_{36})}{s_4} = \frac{(113 + 30)}{1} = 143,$$

and

$$\frac{(MT_5 + t_{36})}{s_5} = \frac{(91 + 30)}{1} = 121,$$

it follows that the job $J_{36}$ is assigned on the processor $M_5$. Thus,

$$ML_5 = \{J_{22}, J_{14}, J_{36}\}, \quad MT_5 = 91 + 30 = 121.$$

Step 3. Choose the job for processing.

Since all jobs in $L_3$ have been assigned, we only need to assign the remaining jobs in $L_1$. Thus the job $J_{15}$ is the candidate. Take $k_1 = 6$.
Step 4. Choose the processor.
Since

$$\frac{(MT_1 + t_{15})}{s_1} = \frac{(102 + 28)}{1.2} = 108.3,$$

$$\frac{(MT_4 + t_{15})}{s_4} = \frac{(113 + 28)}{1} = 141,$$

and

$$\frac{(MT_5 + t_{15})}{s_5} = \frac{(121 + 28)}{1} = 149,$$

it follows that $J_{15}$ is assigned on the processor $M_1$. Thus,

$$ML_1 = \{J_{11}, J_{13}, J_{15}\}, \quad MT_1 = 102 + 28 = 130.$$

Step 3. Choose the job for processing.
Let the job $J_{16}$ be the candidate. Take $k_1 = 7$.
Step 4. Choose the processor.
Since

$$\frac{(MT_1 + t_{16})}{s_1} = \frac{(130 + 22)}{1.2} = 126.7,$$

$$\frac{(MT_4 + t_{16})}{s_4} = \frac{(113 + 22)}{1} = 135,$$

and

$$\frac{(MT_5 + t_{16})}{s_5} = \frac{(121 + 22)}{1} = 143,$$

it follows that the job $J_{16}$ is assigned on the processor $M_1$. Thus,

$$ML_1 = \{J_{11}, J_{13}, J_{15}, J_{16}\}, \quad MT_1 = 130 + 22 = 152.$$

Step 5. If all jobs are assigned, then the program is over.

Up to now, all jobs in any groups have been assigned. So all assigned jobs on each processor and their finish time are the following:

$$ML_1 = \{J_{11}, J_{13}, J_{15}, J_{16}\}, MT_1 = 152, \frac{MT_1}{s_1} = 126.7,$$

$$ML_2 = \{J_{21}, J_{23}, J_{24}\}, MT_2 = 154, \frac{MT_2}{s_2} = 118.5,$$

$$ML_3 = \{J_{31}, J_{32}, J_{34}, J_{35}\}, MT_3 = 180, \frac{MT_3}{s_3} = 120,$$

$$ML_4 = \{J_{12}, J_{33}, J_{25}\}, MT_4 = 113, \frac{MT_4}{s_4} = 113,$$

$$ML_5 = \{J_{22}, J_{14}, J_{36}\}, MT_5 = 121, \frac{MT_5}{s_5} = 121.$$

Thus, $T = 126.7$ and

$$T^* \geq \frac{(T_1 + T_2 + T_3)}{(s_1 + s_2 + s_3 + s_4 + s_5)} = 120.$$

On the other hand, we have the following assignment:

$$ML_1 = \{J_{11}, J_{12}, J_{13}\}, MT_1 = 144, \frac{MT_1}{s_1} = 120,$$

$$ML_2 = \{J_{21}, J_{22}, J_{25}\}, MT_2 = 156, \frac{MT_2}{s_2} = 120,$$

$$ML_3 = \{J_{31}, J_{32}, J_{33}, J_{36}\}, MT_3 = 180, \frac{MT_3}{s_3} = 120,$$

$$ML_4 = \{J_{14}, J_{15}, J_{16}, J_{35}\}, MT_4 = 120, \frac{MT_4}{s_4} = 120,$$

$$ML_5 = \{J_{23}, J_{24}, J_{34}\}, MT_5 = 120, \frac{MT_5}{s_5} = 120.$$

This implies that the optimal solution $T^* = 120$. Thus,

$$\frac{T}{T^*} = \frac{126.7}{120} = 1.0558 < 1 + \frac{2}{(1.2 + 1.3 + 1.5)} = 1.5,$$

which is consistent with the conclusion of Theorem 1.

## References

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

[2] 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.

[3] W. Ding, An improved LS algorithm for the $Q_{m+2}$ $/r_j$ $/C_{max}$ scheduling problem on $m$ general-purpose machineries and two special-purpose machineries, *Comm. On Appl. Math. And Comput.*, **23** (2009), 26–34.

[4] W. Ding, Heuristic algorithm of the $Q//C_{max}$ problem on multi-tasks with uniform processors, *Acta Sci. Natur. Univ. Sunyatseni*, **49** (2010), 5–8.

[5] W. Ding and Y. Zhao, An improved LS algorithm for the problem of scheduling multi groups of jobs on multi processors at the same speed, *Algorithmic Operations Research*, **5** (2010), 34–38.

[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] Cs. 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.