Online hierarchical scheduling: An approach using mathematical programming

Zhiyi Tan *, An Zhang

Department of Mathematics, State Key Lab of CAD & CG, Zhejiang University, Hangzhou 310027, PR China

A R T I C L E   I N F O

Keywords:
Scheduling
Online
Competitive ratio
Mathematical programming
Hierarchy

A B S T R A C T

This paper considers an online hierarchical scheduling problem on parallel identical machines. We are given a set of \( m \) machines and a sequence of jobs. Each machine has a different hierarchy, and each job also has a hierarchy associated with it. A job can be assigned to a machine only if its hierarchy is no less than that of the machine. The objective is to minimize the makespan, i.e., the maximum load of all machines. Two models are studied in the paper. For the fractional model, we present an improved algorithm and lower bounds. Both the algorithm and the lower bound are based on solutions of mathematical programming. For any given \( m \), our algorithm is optimal by numerical calculation. For the integral model, we present both a general algorithm for any \( m \), and an improved algorithm with better competitive ratios of 2.333 and 2.610 for \( m = 4 \) and \( 5 \), respectively.

1. Introduction

In this paper, we consider an online hierarchical scheduling problem on parallel machines. We are given a set of \( m \) parallel machines \([M_1, M_2, \ldots, M_m]\). Machine \( M_i \) has hierarchy \( i \) and speed \( s_i, i = 1, 2, \ldots, m \). A sequence of jobs \( J = \{J_1, J_2, \ldots, J_n\} \) arrives one by one over list. Job \( J_j \) has hierarchy \( g_j \in \{1, 2, \ldots, m\} \) and size \( p_j, j = 1, 2, \ldots, n \). Job \( J_j \) can be assigned to the machine \( M_i \) only if \( g_j \geq i \), and the time needed for \( M_i \) to process \( J_j \) is \( \frac{p_j}{s_i} \). The objective is to minimize the makespan, i.e., the maximum load of all machines. Here the load of a machine is the completion time of any part of the jobs assigned to it.

The performance of an algorithm \( A \) for an online problem is often evaluated by its competitive ratio, which is defined as the smallest number \( \gamma \) such that, for any job sequence \( J \), \( C^A(J) \leq \gamma C^*(J) \), where \( C^A(J) \) (or in short \( C^A \)) denotes the makespan produced by \( A \) and \( C^*(J) \) (or in short \( C^* \)) denotes the optimal makespan in an offline version. An online problem has a lower bound \( \rho \) if no online algorithm has a competitive ratio smaller than \( \rho \). An online algorithm is called \textit{optimal} if its competitive ratio matches the lower bound.

The online hierarchical scheduling problem was first studied by [3], and it has many applications coming from the service industry, computer systems, hierarchical databases, etc. Two models have been studied in the literature. The first is a \textit{fractional} model, where each job can be arbitrarily split between the machines, and parts of the same job can be processed on different machines in parallel. The second is an \textit{integral} model, where each job should be assigned completely to one machine. For the fractional model on \textit{identical} machines, where all machines have the same speed 1, Bar-Noy et al. [3] presented an \( e \)-competitive algorithm. The algorithm can be modified to solve the integral model with competitive ratio \( e + 1 \) (see also [6]). They also showed a lower bound \( e \) for both the fractional model and the integral model when the number of machines tends to infinity.


* Corresponding author; Tel.: +86 571 8795 1429; fax: +86 571 8795 3715.
E-mail address: tanzy@zju.edu.cn (Z. Tan).

0304-3975/– see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2009.08.014
When the number of machines is small, optimal algorithms with better competitive ratios can be obtained, even when the machines have different speeds. Park et al. [11] and Jiang et al. [9] independently presented an optimal algorithm with a competitive ratio of \( \frac{5}{3} \) for the integral model on two identical machines. When the two machines have different speeds, Chassid and Epstein [4] proved that the optimal algorithm has a competitive ratio of \( \frac{1+2+2\sqrt{2}}{1+3+2\sqrt{2}} \) for the fractional model, and Tan and Zhang [12] proved that the optimal algorithm has a competitive ratio of

\[
\begin{align*}
\min \left\{ 1 + s, \frac{2+2s+s^2}{1+s} \right\}, & \quad 0 < s \leq 1, \\
\min \left\{ \frac{1+s}{s}, \frac{1+3+s+s^2}{1+s+s^2} \right\}, & \quad 1 \leq s < \infty,
\end{align*}
\]

for the integral model; here \( s = \frac{m_2}{m_1} \) is the ratio of the speed of the two machines. Recently, Zhang et al. [13] proposed an algorithm with a competitive ratio of 2 for the integral model on three identical machines, and showed that this algorithm is optimal.

A strongly related problem is the online hierarchical scheduling problem on \( m \) identical machines with two hierarchies. In this problem, the machines and jobs have a hierarchy that is either high or low. Jobs with high hierarchy can only be assigned to a machine with high hierarchy, while jobs with low hierarchy can be assigned to all machines. Jiang [10] proposed an online algorithm with a competitive ratio of \( \frac{12+4\sqrt{2}}{7} \approx 2.522 \) for the integral model. The result was improved to \( 1 + \frac{m^2 - m}{m^2 - m_1 + m_2} < \frac{7}{4} \) by Zhang et al. [14], where \( k \) is the number of machines with high hierarchy. They also proved a lower bound of 2 when \( k \geq 3 \) and \( m \geq \left[ \frac{3}{2} (k+1) \right] \). For the fractional model, Zhang et al. [13] proposed an optimal algorithm with competitive ratio \( \frac{m^2}{m^2 - m_1 + m_2} \).

A more general problem is the online restricted assignment model [2], where each job may be processed on a subset of the machines. Azar et al. presented an online algorithm with a competitive ratio of \( \log m + 1 \) for any machine number \( m \). In contrast, if all jobs have the same hierarchy, online hierarchical scheduling reduces to the classical online scheduling problem, which has been well studied during the last half century [8,5,17].

In this paper, we consider both the fractional model and the integral model on identical machines. For the fractional model, we present an online algorithm and lower bounds. Both the algorithm and the lower bounds are based on solutions of mathematical programming. For any given \( m \), our algorithm is better than that in [3] and is optimal by numerical calculation. For the integral model, we first use a method similar to that in [3] to obtain a general algorithm for any \( m \) with better competitive ratio. Then an improved algorithm is given with competitive ratios 2.333 and 2.610 for \( m = 4 \) and 5, respectively.

The rest of the paper is organized as follows. Section 2 gives some useful lemmas. Section 3 considers the online algorithm and lower bounds for the fractional model. Section 4 is dedicated to the integral model. Finally, some concluding remarks are made in Section 5.

2. Preliminaries

Denote by \( T_i \) the total size of jobs with hierarchy \( i, i = 1, \ldots, m \). We have the following results about the optimal makespan of the two models.

**Lemma 2.1** ([3]). For the fractional model, the optimal makespan

\[
C_f^* = \max_{i=1, \ldots, m} \frac{1}{i} \sum_{l=1}^{i} T_l.
\]

For the integral model, the optimal makespan

\[
C_\infty^* \geq \max \left\{ \max_{i=1, \ldots, m} \frac{1}{i} \sum_{l=1}^{i} T_l, \max_{j=1, \ldots, m} \frac{1}{j} \sum_{l=1}^{j} T_l \right\}.
\]

The following technical lemma will be used frequently in the rest of the paper.

**Lemma 2.2.** Let \( 1 = (1, 1, \ldots, 1)^T, x = (x_1, x_2, \ldots, x_q)^T \) be \( q \times 1 \) matrices and \( c = (c_1, c_2, \ldots, c_q) \) be a \( 1 \times q \) matrix. \( A = (a_{ij})_{q \times q} \) is an invertible matrix, and the \( i \)th row vector of \( A \) is denoted as \( a_i \). If \( cA^{-1} \geq 0 \), then

\[
x^Tc \leq (cA^{-1}) \max \{ c_1x, c_2x, \ldots, c_qx \}
\]

for any \( x \).

**Proof.** As \( A \) is invertible, it is clear that \( cx = (cA^{-1})(Ax) \). Moreover, by \( cA^{-1} \geq 0 \), we can conclude that

\[
x^Tc = (cA^{-1})(c_1x, c_2x, \ldots, c_qx)^T \leq (cA^{-1}) \max \{ c_1x, c_2x, \ldots, c_qx \}.
\]

\[\square\]
3. Fractional model

3.1. LP-based algorithm

In this subsection, we give an online algorithm for the fractional model. The algorithm is based on an optimal solution of a linear programming, and performs better than that of Bar-Noy et al. [3] for any given $m$. Besides, the algorithm has very simple structure. The proportion of jobs assigned to each permitted machine only depends on the hierarchy of the job and the machine, regardless of the size of the job.

For any fixed $m \geq 3$, consider the following linear programming:

$$
\begin{align*}
\min & \quad \gamma \\
\text{s.t.} & \quad \sum_{i=1}^{j} x_{ij} = 1, \quad j = 1, \ldots, m, \\
(LP(m)) & \quad ix_{ij} + \sum_{j=i+1}^{m} x_{ij} \leq \gamma, \quad i = 1, \ldots, m, \\
& \quad x_{ij} = 1, \quad j = i, \ldots, m-1, \quad i = 1, \ldots, m, \\
& \quad x_{ij} \geq 0, \quad j = i, \ldots, m, \quad i = 1, \ldots, m.
\end{align*}
$$

Note that $LP(m)$ has $\frac{m^2+m}{2} + 1$ nonnegative decision variables and $\frac{m^2+3m}{2}$ constraints. More importantly, $LP(m)$ only depends on the value of $m$, regardless of the job sequence. So, for fixed $m$, we only need to solve $LP(m)$ once before any job arrives.

Since

$$
\{x_{ij} = 1, i = 1, 2, \ldots, m, x_{ij} = 0, i < j, i, j = 1, 2, \ldots, m, \gamma = m\}
$$

is clearly a feasible solution of $LP(m)$ with objective value $m$, $LP(m)$ always has an optimal solution with bounded objective value. Denote by

$$
\{x_{ij}^{(m)}, i \leq j, i, j = 1, 2, \ldots, m, \gamma^{(m)}\}
$$

the optimal solution of $LP(m)$ with optimal objective value $\gamma^{(m)}$.

**LP algorithm**

For the fractional model of online hierarchical scheduling on $m$ identical machines, when a job $J_k$ with size $p_k$ and hierarchy $g_k = j$ arrives, assign part of $J_k$ with size of $x_{ij}^{(m)}$ to $M_i, i = 1, \ldots, j$.

**Theorem 3.1.** For the fractional model of online hierarchical scheduling on $m$ identical machines, algorithm LP has a competitive ratio of $\gamma^{(m)}$.

**Proof.** Note that (1) ensures the feasibility of $LP$, since parts of $J_k$ are assigned to permitted machines $M_1, \ldots, M_j$, and the total size of parts assigned equals $\sum_{j=1}^{i} x_{ij}^{(m)} p_k = p_k \sum_{j=1}^{i} x_{ij}^{(m)} = p_k$.

Let $\bar{T}_i$ be the load of $M_i$ when the algorithm terminates, $i = 1, \ldots, m$. Since only jobs of hierarchy $j, j \geq i$ will be assigned to $M_i$, and the proportion of jobs assigned to each machine is independent of size, we have

$$
\bar{T}_i = \sum_{j=i}^{m} x_{ij}^{(m)} T_j, \quad i = 1, \ldots, m.
$$

Suppose $C^{LP} = \sum_{i=1}^{T} c_i$ for some $i$. Let $x_{ij} \equiv (T_i, T_{i+1}, \ldots, T_m)^T$ and $c_{ij} \equiv (x_{ij}^{(m)}, x_{i+1}^{(m)}, \ldots, x_{im}^{(m)})$, then $C^{LP} = c_{ij} x_{ij}$. Let

$$
A_{ij} = \begin{pmatrix}
\frac{1}{i} & 0 & 0 & \cdots & 0 \\
\frac{1}{i+1} & \frac{1}{i+1} & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \cdots & \vdots \\
\frac{1}{m} & \frac{1}{m} & \frac{1}{m} & \cdots & \frac{1}{m}
\end{pmatrix}
$$

be an $(m-i+1) \times (m-i+1)$ matrix; the inventory matrix of $A_{ij}$ is

$$
A_{ij}^{-1} = \begin{pmatrix}
\frac{i}{i} & 0 & 0 & \cdots & 0 & 0 \\
-i & \frac{i+1}{i+1} & 0 & \cdots & 0 & 0 \\
0 & -(i+1) & \frac{i+2}{i+2} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \cdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \frac{m-1}{m-1} & 0 \\
0 & 0 & 0 & \cdots & -(m-1) & \frac{m}{m}
\end{pmatrix}.
$$
Since
\[ c_{(i)}A_{(i)}^{-1} = \left( h(x_i^{(m)} - x_{i,i+1}^{(m)}), (i + 1)(x_{i,i+1}^{(m)} - x_{i,i+2}^{(m)}), \ldots, (m - 1)(x_{i,m-1}^{(m)} - x_{i,m}^{(m)}), mx_{i,m}^{(m)} \right), \]
we have \( c_{(i)}A_{(i)}^{-1} \geq 0 \) by constraint (3) of LP(m). On the other hand, by Lemma 2.1, we have
\[ C^*_j = \max_{j=1, \ldots, m} \frac{1}{j} \sum_{i=1}^{j} T_i \geq \max_{j=1, \ldots, m} \frac{1}{j} \sum_{i=1}^{j} T_i = \max_{j=1, \ldots, m+1} \alpha_q x_{(i)}, \]
where \( \alpha_q \) is the jth row vector of \( A_{(i)} \). Hence, by Lemma 2.2, we have
\[ C^{LP} = c_{(i)}A_{(i)}^{-1}1 = \left( c_{(i)}A_{(i)}^{-1}1 \right) \max_{j=1, \ldots, m+1} \alpha_q x_{(i)} \leq \left( c_{(i)}A_{(i)}^{-1}1 \right) C^*_j. \]
Since, for any i,
\[ c_{(i)}A_{(i)}^{-1}1 = i x_i^{(m)} + x_{i,i+1}^{(m)} + \cdots + x_{i,m}^{(m)} \leq \gamma^{(m)}, \]
by constraint (2) of LP(m), we have \( C^{LP} \leq \gamma^{(m)} C^*_j \), and thus the competitive ratio of LP is \( \gamma^{(m)}. \]

3.2. Lower bounds

**Theorem 3.2.** Any algorithm for the fractional model of online hierarchical scheduling on m identical machines has a competitive ratio of at least \( \rho^{(m)} \), where \( \rho^{(m)} \) is the optimal objective value of the following nonlinear programming, NLP(m):

\[
\begin{align*}
\text{max} & \quad \rho \\
\text{subject to} & \quad p_i + q_{i+1} - q_i \geq \rho \max_{j=1, \ldots, m} \frac{1}{j} \sum_{i=1}^{j} p_i, \quad i = 1, \ldots, m, \\
& \quad \rho \geq 1, \quad p_1 = 1, \quad p_i \geq 0, \quad i = 2, \ldots, m, \\
& \quad q_1 = q_{m+1} = 0, \quad q_i \geq 0, \quad i = 2, \ldots, m.
\end{align*}
\]  

**Proof.** Since
\[
\{p_i^{(m)}, i = 1, \ldots, m, q_i^{(m)}, i = 1, \ldots, m + 1, \rho^{(m)} \}
\]
is a feasible solution of NLP(m), the feasible region of NLP(m) will not be empty. Denote by
\[
\{p_i^{(m)}, i = 1, \ldots, m, q_i^{(m)}, i = 1, \ldots, m + 1, \rho^{(m)} \}
\]
an optimal solution of NLP(m). To derive the lower bound for m machines, consider the job sequence with m jobs. The ith job has size \( p_{m-i}^{(m)} \) and hierarchy \( m - i + 1, i = 1, \ldots, m \). By Lemma 2.1, the optimal makespan of the first i jobs is
\[ C^*_i = \max_{k=1, \ldots, m-k+1} \frac{1}{m-k+1} \sum_{l=m-k+1}^{m-1} p_l^{(m)} = \max_{j=i-1, \ldots, m} \frac{1}{j} \sum_{l=i-j+1}^{j} p_l^{(m)}, i = 1, \ldots, m. \]
We will prove by induction that, for any \( 1 \leq i \leq m-1 \), in order to be \( \rho^{(m)} \)-competitive, A must assign parts of the first \( i \) jobs with total size more than \( q_{m-i}^{(m)} \) to the first \( m-i \) machines, \( M_1, \ldots, M_{m-i} \).

If any algorithm A assigns part of the first job with size no more than \( q_{m-i}^{(m)} \) to all machines except \( M_m \), then the sequence terminates. The load of \( M_m \) should be no less than \( p_i^{(m)} - q_{m-i}^{(m)} \). Thus
\[
\frac{C^*_{i} \geq p_{i}^{(m)} - q_{i}^{(m)}}{p_{m}^{(m)} \frac{m}{m}} \geq \rho^{(m)},
\]
where the last inequality is due to (5). The claim is true for \( i = 1 \).

Suppose the claim is true for \( i \), then the \((i+1)\)th job with size \( p_{m-i}^{(m)} \) and hierarchy \( m-i \) arrives. Parts of the first \( i \) jobs that should be assigned to the first \( m-i \) machines, and the \((i+1)\)th job has a total size of at most \( p_{m-i}^{(m)} + q_{m-i+1}^{(m)} \). If the total size assigned to the first \( m-i \) machines \( M_1, \ldots, M_{m-i} \) is no more than \( q_{m-i}^{(m)} \), then the sequence terminates. The load of \( M_{m-i} \) should be no less than \( p_{m-i}^{(m)} + q_{m-i+1}^{(m)} - q_{m-i}^{(m)} \). Thus, by (5) and (6),
\[
C^*_{i} \geq p_{m-i}^{(m)} + q_{m-i+1}^{(m)} - q_{m-i}^{(m)} \geq \rho^{(m)} \max_{j=m-i, \ldots, m} \left( \frac{1}{j} \sum_{l=m-i}^{j} p_l^{(m)} \right) = \rho^{(m)} C^*_j.
\]
For any $j, j$ by 250

Table 1

| Exact and approximation values of the optimal bounds for the fractional model. |
|-----------------|---|---|---|---|---|---|---|
| $m$             | 2 | 3 | 4 | 5 | 6 | 7 |    |
| Exact value     | 1.333 | 1.500 | 1.630 | 1.713 | 1.778 | 1.828 |    |
| Approximation value | 1.869 | 1.905 | 1.936 | 2.106 | 2.265 | 2.351 |    |

LP

respectively. It is interestingtonotethattheoptimalobjectivevaluesof

LP

wehaveexaminedbycomputer,whichimpliesthatalgorithm

3.3. Discussion of the optimality of LP

For any fixed $m \geq 2$, $LP(m)$ and $NLP(m)$ can be solved numerically. For example, the optimal solutions of $LP(4)$ and $NLP(4)$ are

\[
\begin{aligned}
x^{(4)}_{11} &= 1, x^{(4)}_{12} = \frac{14}{27}, x^{(4)}_{13} = \frac{1}{9}, x^{(4)}_{14} = 0, x^{(4)}_{22} = \frac{13}{27}, x^{(4)}_{23} = \frac{5}{27}, x^{(4)}_{33} = x^{(4)}_{34} = \frac{11}{27}, \gamma^{(4)} = \frac{44}{27}
\end{aligned}
\]

and

\[
\begin{aligned}
p^{(4)}_1 = p^{(4)}_2 = p^{(4)}_3 = \frac{3}{2}, p^{(4)}_4 = 1, q^{(4)}_1 = \frac{17}{18}, q^{(4)}_3 = \frac{29}{27}, q^{(4)}_4 = \frac{16}{27}, \rho^{(4)} = \frac{44}{27},
\end{aligned}
\]

respectively. It is interesting to note that the optimal objective values of $LP(m)$ and $NLP(m)$ always coincide for any $m$ that we have examined by computer, which implies that algorithm LP is optimal for these values of $m$. Some selected value of $m$ and the corresponding optimal bounds are listed in Table 1. Therefore, though we neither give an analytical solution of $LP(m)$ or $NLP(m)$, nor prove that the optimal objective values of $LP(m)$ and $NLP(m)$ are identical theoretically, we still conjecture that LP is an optimal algorithm for any $m$.

4. Integral model

4.1. De-fractional LP algorithm

In this subsection, we propose an de-fractional LP algorithm for the integral model. The competitive ratio of the algorithm for the integral model is always larger than that of the fractional model by 1. The idea was first suggested by Bar-Noy et al. [3]. However, our algorithm is based on a more effective LP algorithm for the fractional model, and hence it performs better than that in [3] for any given $m$.

De-fractional LP algorithm (DF for short)

Denote by $l^{k-1}_j$ the current load of machine $M_k$ just before job $j$, arrives, $k = 1, \ldots, m$. Let $L^j_k$ be the load of $M_k$ in the schedule generated by algorithm LP for the job sequence containing first $j$ jobs, $1 \leq k \leq m$. Let

\[
k_j = \max\{k|l^{k-1}_k \leq L^j_k, \text{ for some } 1 \leq k \leq g_j\},
\]

assign $f_j$ entirely to $M_{k_j}$. The feasibility of algorithm DF is based on the following lemma.

Lemma 4.1. For any $j, j = 1, \ldots, n,$

(i) $l^{j-1}_i \leq L^j_i$,

(ii) Let $B^i = \sum_{k=1}^{i} l^j_k$ and $A^i = \sum_{k=1}^{i} L^j_k$, then $B^i \leq A^i$ for any $i = 1, \ldots, m$. 

Proof. We prove the lemma by induction. Clearly $L^0_k = 0$, $k = 1, \ldots, m$. By the description of $LP$, we have

$$
L^1_k = \begin{cases}
x_{k,g_1}p_1, & k = 1, \ldots, g_1, \\
0, & k = g_1 + 1, \ldots, m.
\end{cases}
$$

Therefore, $L^0_k \leq L^1_k$, $1 \leq k \leq m$. Thus (i) is valid for $j = 1$. Clearly, by (7), $k_1 = g_1$ and $DF$ assign $p_1$ to $M_{g_1}$. It follows that $L^1_{g_1} = p_1$ and $L^1_i = 0$, $i \neq g_1$. Hence,

$$
B^1_i = \begin{cases}
0, & i = 1, \ldots, g_1 - 1, \\
p_1, & i = g_1, \ldots, m.
\end{cases}
$$

On the other hand, by (1), we have

$$
A^1_i = \begin{cases}
\sum_{k=1}^{i} L^1_k = p_1 \sum_{k=1}^{i} x_{k,g_1}^{(m)}, & i = 1, \ldots, g_1 - 1, \\
\sum_{k=1}^{g_1} L^1_k = p_1 \sum_{k=1}^{g_1} x_{k,g_1}^{(m)} = p_1, & i = g_1, \ldots, m.
\end{cases}
$$

Hence, (ii) is also true for $j = 1$.

Now suppose that (i) and (ii) hold for $j$, and consider the case of $j + 1$. First, we have $L^{j+1}_1 = A^{j+1}_1 \geq A^j_1 \geq B^j_1 = L^j_1$ by inductive assumption, which implies that (i) still holds for $j + 1$. Hence, $k_{j+1}$ exists and $k_{j+1} \leq g_{j+1}$. Note that $L^{j+1}_{g_{j+1}} = L^{j+1} + p_{j+1}$ and $L^{j+1}_{g_{j+1}} = L^{j+1}_{g_{j+1}+1}$, $k \neq k_{j+1}$. Hence,

$$
B^{j+1}_i = \begin{cases}
B^j_i, & i = 1, \ldots, k_{j+1} - 1, \\
B^j_i + p_{j+1}, & i = k_{j+1}, \ldots, m.
\end{cases}
$$

On the other hand, by the description of $LP$, we have

$$
L^{j+1}_k = \begin{cases}
L^j_k + x_{k,g_{j+1}}^{(m)}p_{j+1}, & k = 1, \ldots, g_{j+1}, \\
L^j_k, & k = g_{j+1} + 1, \ldots, m.
\end{cases}
$$

It follows that

$$
A^{j+1}_i = \begin{cases}
A^j_i + \sum_{k=1}^{i} x_{k,g_{j+1}}^{(m)}p_{j+1}, & i = 1, \ldots, g_{j+1} - 1, \\
A^j_i + \sum_{k=1}^{g_{j+1}} x_{k,g_{j+1}}^{(m)}p_{j+1} = A^j_g + p_{j+1}, & i = g_{j+1}, \ldots, m.
\end{cases}
$$

If $k_{j+1} = g_{j+1}$, then by the inductive assumption $B^j_i \leq A^j_i$ for $i = 1, \ldots, m$, we also have $B^{j+1}_i \leq A^{j+1}_i$ for $i = 1, \ldots, m$. Otherwise, $k_{j+1} < g_{j+1}$. Clearly, $B^{j+1}_i \leq A^{j+1}_i$ is still valid for $1 \leq i \leq k_{j+1} - 1$ and $g_{j+1} \leq i \leq m$. For $k_{j+1} \leq i \leq g_{j+1} - 1$, we must have $L^j_k > L^{j+1}_k$ by (7). Hence,

$$
A^{j+1}_i = \sum_{k=1}^{i} L^{j+1}_k = \sum_{k=1}^{g_{j+1}} L^{j+1}_k - \sum_{k=1}^{g_{j+1}} L^{j+1}_k = A^{j+1}_{g_{j+1}} - \sum_{k=1}^{g_{j+1}} L^{j+1}_k
$$

(ii) also holds for $j + 1$.

Theorem 4.1. For the integral model of the online hierarchical scheduling problem on $m$ identical machines, algorithm $DF$ has a competitive ratio of $\gamma^{(m)} + 1$.

Proof. Note that (i) of Lemma 4.1 implies that $k_j$ always exists for any $j$, then $DF$ is well-defined. W.l.o.g., suppose that the last job $f_g$ determines the makespan. Hence, $C^{DF} = L^{m-1}_{k_m} + p_n$. By the definition of $k_n$, we have $L^{m-1}_{k_n} \leq L^m_{k_n}$. Combining with Theorem 3.1 and Lemma 2.1, it follows that

$$
C^{DF} \leq L^m_{k_n} + p_n \leq C^{LP} + p_n \leq \gamma^* C^*_g + p_n \leq (\gamma^{(m)} + 1) C^*_g.
$$

From Table 1 and Theorem 4.1, we know that the competitive ratios of $DF$ for $m = 2, 3$ machines are 7/3 and 5/2, respectively.

4.2. Improved algorithm for small $m$

Though $LP$ has good performance for the fractional model, the competitive ratio of $DF$ seems a bit large, especially for small $m$. In fact, current results show that optimal algorithms for the integral model of online hierarchical scheduling on two
and three machines have competitive ratios of 5/3 and 2, respectively [11,9,13], both smaller than those of DF. Hence, in this subsection, we propose an improved algorithm which has better performance than DF for \( m = 4, 5 \). Though the algorithm can be used for situations with more machines as well, the analysis seems to become extremely troublesome.

We first generalize the result of Lemma 2.1. Denote \( V_\beta = \{ k | 1 \leq k \leq j \text{ and } g_k = i \} \) and \( T_\beta = \sum_{k \in V_\beta} p_k \) for any \( i = 1, \ldots, m \). Let

\[
LB_j = \max \left\{ \max_{i=1, \ldots, m} \frac{1}{i} \sum_{l=1}^{i} T_{\beta}, \max_{i=1, \ldots, j} p_i \right\}.
\]

Clearly, \( LB_j \) is a nondecreasing function of \( j \).

**Lemma 4.2.** For the job sequence containing the first \( j \) jobs, \( J_1, J_2, \ldots, J_j \), the optimal makespan of the integral model is at least \( LB_j \).

**Hierarchical threshold \((m; \alpha)\) algorithm \((HT\) for short)\:**

Let \( J_j \) be the currently arriving job, and \( L_{j-1} \) be the current load of \( M_i \) just before \( J_j \) arrives, \( i = 1, \ldots, m \).

1. If \( g_j = 1 \text{ or } g_j = 2 \), assign \( J_j \) to \( M_i \).
2. If \( g_j \geq 4 \), then let

\[
I_j = \{ l | p_l + L_{j-1} \leq (1 + \alpha)LB_j \text{ and } 4 \leq l \leq g_j \}.
\]

If \( I_j \neq \emptyset \), assign \( J_j \) to \( M_k \), where \( k = \arg \max I_j \). Otherwise, goto Step 3.

3. If \( L_{j-1} < L_3 \), assign \( J_j \) to \( M_k \). Otherwise, assign \( J_j \) to \( M_3 \).

In fact, \( HT \) uses three different methods according to the hierarchy of the jobs. \( M_1 \) is specialized for processing jobs of hierarchy 1 or 2. The load of \( M_1 \) will not be too large since the expected competitive ratio is greater than 2. The assignment of jobs with hierarchy no less than 4 is according to the dual greedy idea. Jobs are assigned to the machine with hierarchy as large as possible, unless such assignment will either be not permitted due to a hierarchy constraint, or will cause the load of that machine to exceed a certain threshold, which is determined by Lemma 4.2. Since jobs with hierarchy 2 are already scheduled on \( M_1 \), we use \( M_2 \) and \( M_3 \) to process jobs with hierarchy 3, and all remaining jobs which cannot be assigned to machines with larger hierarchy. Since no hierarchy constraints should be involved now, we assign these jobs using the primal greedy idea. As we will see in the following, the combination of these methods makes the loads of all machines as even as can be, and greatly simplifies the case by case analysis.

**Theorem 4.2.** For \( m = 4 \) and \( \alpha = \frac{3}{4} \), \( HT(4; \alpha) \) is no more than \((1 + \alpha)\)-competitive.

**Proof.** W.l.o.g., suppose that the last job \( J_n \) determines the makespan. If it is assigned to \( M_4 \), then clearly we have \( C^{HT} = p_n + L_4^n = (1 + \alpha)LB_n \leq (1 + \alpha)C^*_g \), where the last inequality is due to Lemma 2.1. If the algorithm assigns \( J_n \) to \( M_1 \), then we have

\[
C^{HT} = L_1^n = T_{n1} + T_{n2} = \frac{T_{n1} + T_{n2}}{2} \leq 2C^*_g,
\]

since the jobs assigned to \( M_1 \) are of hierarchy either 1 or 2. Now suppose that \( J_n \) is assigned to \( M_2 \) or \( M_3 \) by the algorithm. Since \( HT \) always assigns jobs to the one of the two machines which has the smaller load, we have

\[
C^{HT} \leq \frac{L_2^{n-1} + L_3^{n-1}}{2} + p_n.
\]

If there are only jobs with hierarchy 3 assigned to \( M_2 \) and \( M_3 \), then we must have \( T_{n3} \geq L_2^{n-1} + L_3^{n-1} + p_n \). Together with Lemma 2.1, it follows that

\[
C^*_g \geq \max \left\{ \sum_{i=1}^{3} T_{ni} \right\} \geq \max \left\{ \frac{L_2^{n-1} + L_3^{n-1} + p_n}{3}, p_n \right\}.
\]

Let \( c = (1, 1), x = (\frac{L_2^{n-1} + L_3^{n-1}}{2}, p_n) \) and \( A = \left( \frac{3}{2}, \frac{1}{3} \right) \). By (9), it is easy to verify that

\[
C^{A} = c x \geq \max(\alpha_1 x, \alpha_2 x), \quad c A^{-1} = (1, 1) \begin{pmatrix} \frac{3}{2} & \frac{1}{3} \\ \frac{3}{2} & \frac{1}{2} \end{pmatrix} = \begin{pmatrix} 3 & 1 \\ 2 & 1 \end{pmatrix} \geq 0.
\]

By Lemma 2.2, we can get \( C^{HT} \leq c A^{-1} 1 C^*_g = 2C^*_g < \frac{7}{3} C^*_g \).

Therefore, we assume that there exist jobs with hierarchy 4 assigned to \( M_2 \) or \( M_3 \) in the following. Denote by \( J_j \) the last job of the sequence which is assigned to \( M_2 \) or \( M_3 \) and does not have hierarchy 3. In other words, each job assigned to \( M_2 \) and \( M_3 \) and arriving after \( J_j \) must be of hierarchy 3. Hence, due to Lemma 2.1, one obtains

\[
C^*_g \geq \frac{\sum_{i=1}^{3} T_{ni}}{3} \geq \frac{T_{n3}}{3} \geq \frac{1}{3} (L_2^{n-1} + L_3^{n-1} + p_n) - \frac{1}{3} (L_2^{j-1} + L_3^{j-1} + p_j).
\]

\[\text{Lemma 2.1.}\]

\[\text{Theorem 4.2.}\]
Since $J_1$ cannot be assigned to $M_4$ by the algorithm, $L_i^{-1} + p_j > (1 + \alpha)LB_j$, and thus $L_i^{-1} > \alpha LB_j$ by $LB_j \geq p_j$. By the definition of $L_B$ and $\alpha = \frac{4}{7}$, we have
\[
L_i^{-1} + L_j^{-1} + p_i \leq T_j - L_i^{-1} \leq 4LB_j - L_i^{-1} < (4 - \alpha)LB_j = 2\alpha LB_j < 2L_i^{-1}.
\]
It follows directly that
\[
\sum_{i=1}^{4} T_{ni} \geq L_2^{-1} + L_3^{-1} + p_n + \frac{L_i^{-1} + L_i^{-1} + p_i}{2}.
\]
(11)

For simplicity, let $x_1 = \alpha x_2 = (L_2^{-1} + L_3^{-1}) - (L_i^{-1} + L_i^{-1} + p_i)$ and $x_3 = (L_i^{-1} + L_i^{-1} + p_i)$; then combining with Lemma 2.1 and (10) and (11), we can conclude that
\[
\begin{aligned}
C^*_g \geq & \max \left\{ x_1, \frac{x_1 + x_2}{3}, \frac{x_1 + x_2 + \frac{3}{2}x_3}{4} \right\}.
\end{aligned}
\]
(12)

Now, let $c = (1, \frac{1}{2}, \frac{1}{2})$, $x = (x_1, x_2, x_3)^T$ and $A = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$; it can be verified from (9) and (12) that $C^{HT} \leq cx$.

Thus, by Lemma 2.2, we have
\[
C^{HT} \leq cA^{-1}c^* = \frac{7}{2}c^* \leq (1 + \alpha)c^*.
\]
\[
\Box
\]

Theorem 4.3. For $m = 5$ and $\alpha \approx 1.610$, which is the smaller positive solution of equation $2x^3 + 11x^2 - 85x + 100 = 0$, $HT(5; \alpha)$ is no more than $(1 + \alpha)$-competitive.

Proof. W.l.o.g., suppose that the last job $J_5$ determines the makespan. If it is assigned to $M_5$ or $M_4$, then clearly we have $C^{HT} = p_n + L_n^{-1} \leq (1 + \alpha)LB_n \leq (1 + \alpha)c^*_g$, $i = 4, 5$, where the last inequality is due to Lemma 2.1. If the algorithm assigns $J_6$ to $M_1$, then we have
\[
C^{HT} = L_i^n = T_{n1} + T_{n2} = 2 \frac{T_{n1} + T_{n2}}{2} \leq 2c^*.
\]
since the jobs assigned to $M_1$ are of hierarchy either 1 or 2. Now suppose that $J_6$ is assigned to $M_2$ or $M_3$ by the algorithm. Since $HT$ always assigns jobs to the one of the two machines which has the smaller load, we have
\[
C^{HT} \leq L_2^{-1} + L_3^{-1} + p_n.
\]
(13)

If there are only jobs with hierarchy 3 assigned to $M_2$ and $M_3$, then $T_{n2} \geq L_2^{-1} + L_3^{-1} + p_n$. Together with Lemma 2.1, it follows that
\[
C^*_g \geq \max \left\{ \frac{\sum_{i=1}^{3} T_{ni}}{3}, p_n \right\} \geq \max \left\{ \frac{T_2^{-1} + T_3^{-1} + p_n}{4}, p_n \right\}.
\]

Let $c = (1, 1)$, $x = (\frac{T_2^{-1} + T_3^{-1}}{2}, p_n)^T$ and $A = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{bmatrix}$. By (13), it is easy to verify that
\[
C^* \leq cx, \quad C^* \geq \max \{\alpha_1 x, \alpha_2 x\}, \quad cA^{-1} = (1, 1) \begin{bmatrix} 2 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & 1 \end{bmatrix} \geq 0.
\]

By Lemma 2.2, we can get $C^{HT} \leq cA^{-1}c^* = \frac{5}{2}c^* < (1 + \alpha)c^*$.

Therefore, we assume that there exist jobs with hierarchy greater than 3 assigned to $M_2$ or $M_3$. In the following, denote by $J_i$ the last job of the sequence which is assigned to $M_2$ or $M_3$ and does not have hierarchy 3. In other words, each job assigned to $M_2$ and $M_3$ and arriving after $J_i$ must be of hierarchy 3. Hence, due to Lemma 2.1,
\[
C^*_g \geq \frac{\sum_{i=1}^{3} T_{ni}}{3} \geq \frac{T_{n2}}{3} \geq \frac{1}{3}(L_2^{-1} + L_3^{-1} + p_n) - \frac{1}{3}(L_2^{-1} + L_3^{-1} + p_i).
\]
(14)

We distinguish two cases according to the value of $g_i$. 

Case 1. $g_j = 5$.

Since $j_1$ cannot be assigned to $M_4$ or $M_5$ by the algorithm, we must have $\mu_{i}^{j-1} + p_j > (1+\alpha)LB_j$ and $\mu_{i}^{j-1} + p_j > (1+\alpha)LB_j$. Hence, $\mu_{i}^{j-1} > \alpha LB_j$, $\mu_{i}^{j-1} > \alpha LB_j$ and

$$
LB_j \geq \frac{\sum_{i=1}^{5} T_{ji}}{5} = \frac{\sum_{i=1}^{5} \mu_{i}^{j-1} + p_j}{5} \geq \frac{(\mu_{2}^{j-1} + \mu_{3}^{j-1} + p_j) + (\mu_{4}^{j-1} + \mu_{5}^{j-1})}{5}
$$

by (8). It follows that $\mu_{4}^{j-1} + \mu_{5}^{j-1} > 2\alpha LB_j$ and thus

$$
\mu_{2}^{j-1} + \mu_{3}^{j-1} + p_j \leq (5 - 2\alpha)LB_j < \frac{5 - 2\alpha}{2\alpha}(\mu_{4}^{j-1} + \mu_{5}^{j-1}).
$$

By Lemma 2.1, we have

$$
C^*_g \geq \frac{\sum_{i=1}^{5} T_{ni}}{5} \geq \frac{(\mu_{2}^{n-1} + \mu_{3}^{n-1}) + (\mu_{4}^{n-1} + \mu_{5}^{n-1}) + p_n}{5} \geq \frac{1}{5} p_n + \frac{1}{5} (\mu_{2}^{n-1} + \mu_{3}^{n-1}) + \frac{2\alpha}{5(5 - 2\alpha)}(\mu_{2}^{j-1} + \mu_{3}^{j-1} + p_j).
$$

Let $y_1 = p_n, y_2 = (\mu_{2}^{n-1} + \mu_{3}^{n-1}) - (\mu_{2}^{j-1} + \mu_{3}^{j-1} + p_j)$ and $y_3 = (\mu_{2}^{j-1} + \mu_{3}^{j-1} + p_j)$. Combining with Lemma 2.1 and (14) and (15), we have

$$
C^*_g \geq \max \left\{ y_1, \frac{1}{3} y_1 + \frac{1}{3} y_2, \frac{1}{5} y_1 + \frac{1}{5} y_2 + \frac{1}{5 - 2\alpha} y_3 \right\}.
$$

Let $c = (1, \frac{1}{3}, \frac{1}{2})$, $x = (y_1, y_2, y_3)^T$ and $A = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$; it can be verified from (13) and (16) that $c^{HT} \leq cx).

$$
C^*_g \geq \max \{\alpha_1 x, \alpha_2 x, \alpha_3 x\}
$$

Thus, by Lemma 2.2, we have $C^{HT} \leq cA^{-1}C^*_g = (3 - \frac{3}{2}\alpha)C^*_g < (1 + \alpha)C^*_g$.

Case 2. $g_j = 4$.

By algorithm $HT$ and the definition of $LB_j$, it follows that $p_j + \mu_{4}^{j-1} > (1+\alpha)LB_j \geq p_j + \alpha LB_j$ and

$$
LB_j \geq \frac{\sum_{i=1}^{5} T_{ji}}{5} \geq \frac{\mu_{2}^{j-1} + \mu_{3}^{j-1} + \mu_{4}^{j-1} + p_j}{5} \geq \frac{\mu_{2}^{j-1} + \mu_{3}^{j-1} + \alpha LB_j + p_j}{5},
$$

which lead to

$$
\mu_{4}^{j-1} > \alpha LB_j > \frac{\alpha}{5 - \alpha}(\mu_{2}^{j-1} + \mu_{3}^{j-1} + p_j).
$$

If no job assigned to $M_2$ or $M_3$ before the arrival of $j_1$ has hierarchy 5, in other words, all jobs assigned to $M_2$ and $M_3$ are of hierarchy 3 or 4, we must have $T_{n3} + T_{n4} \geq \mu_{2}^{n-1} + \mu_{3}^{n-1} + p_n$ and

$$
C^*_g \geq \max \left\{ \frac{\sum_{i=1}^{4} T_{ni}}{4} \right\} \geq \max \left\{ p_n, \frac{\mu_{2}^{n-1} + \mu_{3}^{n-1} + p_n}{4} \right\},
$$

by Lemma 2.1. By the same analysis as before, we can get $C^{HT} \leq \frac{5}{2} C^*_g$.

Hence, suppose there are some jobs with hierarchy 5 which arrive before $j_1$ and are assigned to $M_2$ or $M_3$. Let $j_k$ be the one among them which arrived last. In other words, all jobs assigned to $M_2$ or $M_3$ arriving after $j_k$ must have hierarchy 3 or 4. Thus we have

$$
C^*_g \geq \frac{\sum_{i=1}^{4} T_{ni}}{4} \geq \frac{(\mu_{2}^{n-1} + \mu_{3}^{n-1} + p_n) - (\mu_{k}^{k-1} + \mu_{k}^{k-1} + p_k)}{4}.
$$
Note that the algorithm prefers to assign jobs with hierarchy 5 to $M_5$ or $M_6$; it follows that $p_k + L_5^{k-1} > (1 + \alpha)LB_k \geq \alpha LB_k + p_k$ and $p_k + L_4^{k-1} > (1 + \alpha)LB_k \geq \alpha LB_k + p_k$. Hence,

$$LB_k \geq \frac{1}{5} \left( \frac{(L_4^{k-1} + L_3^{k-1} + p_k) + (L_4^{k-1} + L_5^{k-1})}{5} \right) \geq \frac{(L_2^{k-1} + L_3^{k-1} + p_k) + 2\alpha LB_k}{5},$$

which further results in $L_5^{k-1} > \alpha LB_k \geq \frac{\alpha}{5 - 2\alpha} (p_k + L_2^{k-1} + L_3^{k-1})$. Combining this with Lemma 2.1 and (17), we have

$$C^* \geq \frac{1}{5} \left( \frac{p_n + L_n^{n-1} + L_4^{n-1} + L_5^{n-1}}{5} \right) \geq \frac{p_n + L_2^{n-1} + L_3^{n-1}}{5} + \alpha \frac{p_k + L_2^{k-1} + L_3^{k-1}}{5} - \frac{5\alpha - \alpha^2}{5(5 - 2\alpha)} z_4.$$

Let $z_1 = p_n, z_2 = (L_2^{n-1} + L_3^{n-1})$, $z_3 = (p_j + L_2^{j-1} + L_3^{j-1}) - (p_k + L_2^{k-1} + L_3^{k-1})$ and $z_4 = p_k + L_2^{k-1} + L_3^{k-1}$. By Lemma 2.1 and (14), (18) and (19), we can conclude that

$$C^* \geq \max \left\{ z_1, \frac{z_1 + z_2}{3}, \frac{z_1 + z_2 + z_3}{4}, \frac{z_1 + z_2 + z_3 + z_4}{5}, \frac{25 - 5\alpha - \alpha^2}{5(5 - 2\alpha)} \right\}.$$

Let

$$c = \begin{pmatrix} 1 & 1 \frac{1}{2} & 1 \frac{1}{2} \end{pmatrix}, \quad x = (z_1, z_2, z_3, z_4)^T, \quad A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{3}{5} & \frac{1}{3} & 0 \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{3} \end{pmatrix},$$

it can be verified from (13) and (20) that $C^{HT} \leq cx, C^* \geq \max \{ \alpha_1 x, \alpha_2 x, \alpha_3 x, \alpha_4 x \}$ and

$$cA^{-1} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 3 & 0 & 0 \\ 0 & -3 & 4 & 0 \\ 15\alpha - 6\alpha^2 & 2\alpha(5 - \alpha) & 125 - 75\alpha + 10\alpha^2 & 25 - 5\alpha - \alpha^2 \end{pmatrix} \begin{pmatrix} \frac{1}{2} & \frac{3\alpha(5 - 2\alpha)}{50 - 10\alpha - 2\alpha^2} \\ \frac{1}{2} & \frac{2\alpha(5 - \alpha)}{25 - 5\alpha - \alpha^2} \\ \frac{1}{2} & \frac{125 - 75\alpha + 10\alpha^2}{25 - 5\alpha - \alpha^2} \end{pmatrix} \geq 0.$$

Thus, by Lemma 2.2, we have $C^{HT} \leq cA^{-1}c = \frac{150 - 44\alpha - \alpha^2}{50 - 10\alpha - 2\alpha^2} C^* = (1 + \alpha)C^*_g$, where the last equality holds by the definition of $\alpha$. \hfill \Box

### 4.3. General hierarchy setting

During the above analysis we assume that there are exactly $m$ hierarchies among $m$ machines, which is the maximum number of different hierarchies that can be set. However, two or more machines can share one hierarchy in some real applications. In fact, the problem with general hierarchy setting can be viewed as a subproblem of the problem with maximum number of hierarchies. In detail, suppose that there are $k < m$ hierarchies among $m$ machines; the number of machines with hierarchy $i$ is $m_i, i = 1, \ldots, k$. Clearly, any job $j$ in sequence $g$ for the $k$ hierarchy problem has hierarchy $g_j \in \{1, 2, \ldots, k\}$. We construct a sequence $g'$ for the $m$ hierarchy problem from $g$ as follows. The job with hierarchy $i$ in $g$ has hierarchy $\sum_{i=1}^{j-1} m_i$ in $g'$. The size of each job remains unchanged. The two problems are equivalent since a one-to-one mapping between the two machine sets can be easily made such that the permitted machines of each job will remain unchanged. Hence, the competitive ratio of the optimal algorithm for problems with any hierarchy setting will not be greater than the optimal algorithm for the problem with maximum number of hierarchies. Table 2 provides a comparison on the competitive ratios of algorithms for some typical cases, including classical online scheduling (one hierarchy) and two-hierarchy scheduling. However, to design optimal algorithms for each hierarchy setting is complicated, unless some general method can be found. It still remains an open question whether the competitive ratio of the optimal algorithm for a problem with maximum number of hierarchies will be strictly larger than that for a problem with any other hierarchy setting.

### 5. Concluding remarks

In this paper, we have studied online hierarchical scheduling problems on $m$ parallel identical machines. Both the fractional model and the integral model are considered. For the fractional model, we present an improved algorithm and
lower bounds. For any given m, our algorithm is optimal by numerical calculation. For the integral model, we present both a general algorithm for any m, and an improved algorithm with better competitive ratios for small m.

The results of this paper suggest a number of problems deserving further study. Perhaps the most interesting one is to prove that the optimal objective values of LP(m) and NLP(m) are identical, or to obtain optimal solutions of LP(m) or NLP(m) with respect to m. Secondly, the algorithm HT is designed for problems with any number of machines, but only competitive ratios for small m are obtained. Is its competitive ratio still less than that of DF when m becomes large? We conjecture that some adjustment of the algorithm should be made. Finally, little is known about the lower bounds of the integral model when m ≥ 4. Obviously, the lower bound of the fractional model is also a lower bound of the corresponding integral model, and the sequence used to prove the lower bound 2 for m = 3 [13] can be adopted to the case of m > 3 as well. But can we get better lower bounds for m ≥ 4 machines? This plays an important role in obtaining optimal algorithms.

Acknowledgement

The authors wish to thank Dr. Gy. Dósa for his valuable discussion, and two anonymous referees for their valuable comments.

The first author was supported by the National Natural Science Foundation of China (10971191, 60021201) and Zhejiang Provincial Natural Science Foundation of China (Y607079).

References