

Online Scheduling of Parallel Jobs on Two Machines is 2-Competitive

J.L. Hurink and J.J. Paulus*

University of Twente, P.O. box 217, 7500AE Enschede, The Netherlands

Abstract

We consider online scheduling of parallel jobs on parallel machines. For the problem with two machines and the objective of minimizing the makespan, $P2|online - list, m_j|C_{max}$, we show that 2 is a lower bound on the competitive ratio of any online algorithm. Thereby we not only improve the existing lower bound of $1 + \sqrt{\frac{2}{3}}$, but also close the gap with the trivial upper bound of 2. For the problem with m machines, $Pm|online - list, m_j|C_{max}$, we derive lower bounds using an ILP formulation. Furthermore, we show that with the presented construction no lower bound greater than 2.5 can be obtained.

Key words: Online Machine Scheduling, Parallel Jobs, Competitive Analysis

1 Introduction

In recent years the problem of scheduling parallel jobs on parallel machines gained considerable attention. Contrary to classical parallel machine scheduling problems, jobs may require processing on several machines in parallel. Applications, like computer architectures with parallel processors, motivate the study of these type of scheduling problems. For an overview of recent developments on this type of scheduling problems see Johannes [4].

In this paper we study the problem of online scheduling of parallel jobs on two parallel machines. Jobs are presented one by one to the decisionmaker, and are characterized by their processing time and the number of machines simultaneously required for processing (1 or 2 machines). As soon as a job gets known, it has to be scheduled (i.e. its start time has to be set) irrevocably without knowing the characteristics of the future jobs. Preemption

* Corresponding author. e-mail: j.j.paulus@ewi.utwente.nl

is not allowed and the objective is to minimize the makespan. Adopting the notation from Pruhs et al. [5] and Johannes [4], this problem is denoted by $P2|online - list, m_j|C_{max}$. In this paper we show that no online algorithm for this problem can have competitive ratio strictly less than 2.

For the evaluation of an online algorithm ON , competitive analysis is used. For any sequence σ of jobs we compare the makespan of the schedule generated by the online algorithm $C_{ON}(\sigma)$ with the makespan of the optimal offline solution $C_{OPT}(\sigma)$. An online algorithm is said to be ρ -competitive if $\sup_{\sigma} \frac{C_{ON}(\sigma)}{C_{OPT}(\sigma)} \leq \rho$. For background information on online algorithms, see e.g. Albers [1] and Borodin and El-Yaniv [2], and on online scheduling, see e.g. Pruhs et al. [5].

The online scheduling of parallel jobs on two machines has previously been studied by Chan et al. [3]. They proved a lower bound of $1 + \sqrt{\frac{2}{3}}$ on the competitive ratio of any online algorithm. For the case where jobs arrive in non-decreasing order of processing time, they give an optimal $\frac{3}{2}$ -competitive algorithm. And for the case where jobs arrive in non-increasing order of processing, they give a $\frac{4}{3}$ -competitive algorithm and a lower bound of $\frac{9}{7}$ on the competitive ratio of any online algorithm. For the general problem, with an arbitrary number of machines, $P|online - list, m_j|C_{max}$, Johannes [4] was the first to develop an online algorithm with constant competitive ratio. She gave a 12-competitive online algorithm, which was later improved by Ye and Zhang [6] to an 8-competitive algorithm. For the problem with two machines, a greedy algorithm which schedules the jobs upon arrival as early as possible, has a competitive ratio of at most 2. This follows directly from the fact that never both machines are left idle by such a greedy algorithm.

In Sections 2 and 3, we prove that no online algorithm can have a competitive ratio strictly less than 2. We construct a series of job sequences in which jobs have an alternate machine requirement of 1 and 2, and show that no online algorithm can have competitive ratio strictly less than 2 for these sequences. Therefore, the greedy algorithm is the best possible for the considered online problem with two machines. In Section 4, we derive lower bounds for $Pm|online - list, m_j|C_{max}$ using an ILP formulation. We show the limitation of the instance construction, by proving that no lower bound greater than 2.5 can be obtained with that type of instance.

2 Lower Bound on the Competitive Ratio

To prove a lower bound of 2 on the competitive ratio of any online algorithm for $P2|online - list, m_j|C_{max}$, we are going to construct a series of job sequences and argue that no online algorithm can have a makespan strictly less than twice the makespan of the optimal offline solution. In the following we assume

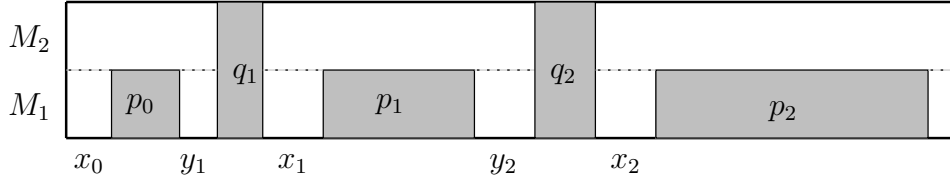


Fig. 1. Structure of the online schedule for σ_2

ON to be an online algorithm with competitive ratio strictly less than 2, and we show that such an algorithm cannot exist. By $C_{OPT}(\sigma)$ and $C_{ON}(\sigma)$ we denote the makespan of the optimal offline schedule and the makespan of the schedule constructed by the online algorithm ON on the job sequence σ , respectively.

We define σ_n as the sequence of jobs $(p_0, q_1, p_1, q_2, p_2, \dots, q_n, p_n)$, where p_i (q_i) denotes a job with processing time p_i (q_i) and a machine requirement of 1 (2). The job lengths are defined as:

$$\begin{aligned}
 p_0 &= 1 \\
 p_1 &= \max\{2, x_0 + p_0 + y_1 + \epsilon\} \\
 p_i &= 2 \cdot p_{i-1} \quad \forall i \geq 2 \\
 q_1 &= x_0 + \epsilon \\
 q_i &= \max\{y_{i-1}, q_{i-1}, x_{i-1}\} + \epsilon \quad \forall i \geq 2,
 \end{aligned}$$

where x_i and y_i are values given by delays the online algorithm has used for placing earlier jobs. Therefore, the job lengths are depending on the online algorithm ON . The concrete definition of these values is given in the next paragraph.

In Lemma 1 we prove that any online algorithm with competitive ratio strictly less than two has to schedule the jobs in the same order as they appear in the sequence σ_n . As a consequence, Figure 1 illustrates the structure of the online schedule. Therefore, the only remaining decision for the online algorithm ON is to decide how long it delays the start of a job, i.e. how much time is left between the start of the current job and the completion of the previous job. We denote by x_i (y_i) the delay incurred by ON on job p_i (q_i), completing thereby also the definition of the processing times q_i .

To simplify notation for the remaining, we let ϵ go to zero and omit it from the rest of the analysis. Using the result of Lemma 1, the structure of the online schedule for σ_n is fixed and its makespan is given by:

$$C_{ON}(\sigma_n) = x_0 + p_0 + \sum_{i=1}^n (y_i + q_i + x_i + p_i)$$

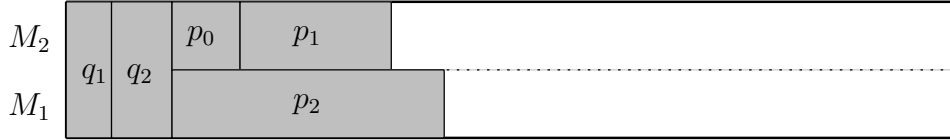


Fig. 2. Structure of the optimal offline schedule for σ_2

The optimal schedule for σ_n is obtained by scheduling the jobs p_0, \dots, p_{n-1} parallel to job p_n after a block containing the jobs q_1, \dots, q_n (see Figure 2). Therefore, the makespan of the optimal schedule is given by:

$$C_{OPT}(\sigma_n) = \sum_{i=1}^n q_i + p_n$$

Using these makespans for the job sequence σ_n , we can calculate the competitive ratio of online algorithm ON on this particular instance. Note that $p_n = 2^{n-1} \cdot p_1$ and $\sum_{i=1}^n p_i = (2^n - 1)p_1$

$$\begin{aligned} \frac{C_{ON}(\sigma_n)}{C_{OPT}(\sigma_n)} &= \frac{x_0 + p_0 + \sum_{i=1}^n (y_i + q_i + x_i + p_i)}{\sum_{i=1}^n q_i + p_n} \\ &= \frac{x_0 + p_0 + (2^n - 1) \cdot p_1 + \sum_{i=1}^n (y_i + q_i + x_i)}{\sum_{i=1}^n q_i + 2^{n-1} \cdot p_1} \\ &= 2 - \frac{\sum_{i=1}^n q_i - \sum_{i=1}^n (y_i + x_i) - x_0 - p_0 + p_1}{\sum_{i=1}^n q_i + 2^{n-1} \cdot p_1} \end{aligned}$$

In Lemma 2 we prove that

$$\frac{\sum_{i=1}^n q_i - \sum_{i=1}^n (y_i + x_i) - x_0 - p_0 + p_1}{\sum_{i=1}^n q_i + 2^{n-1} \cdot p_1} \rightarrow 0$$

as n goes to infinity for any online algorithm with competitive ratio strictly less than 2. However, this is a contradiction with the competitive ratio being strictly less than 2. As a result, we have proven our main theorem:

Theorem 1 *No online algorithm for $P2|online - list, m_j|C_{max}$ has a competitive ratio strictly less than 2.* \square

It remains to prove the two lemmata.

3 Proof of the Lemmata

Lemma 1 *If an online algorithm ON has a competitive ratio strictly less than 2, it can only schedule the jobs in the same order as they appear in σ_n .*

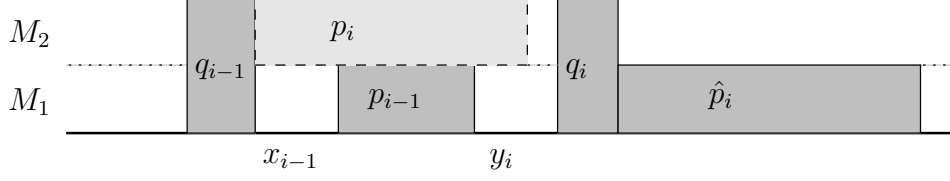


Fig. 3. Present the online algorithm with an alternative job \hat{p}_i

Proof: By definition of the length of job q_i there is no gap in the current schedule before p_{i-1} in which job q_i can be scheduled. The same for holds for p_1 . Thus, it only remains to prove that job p_i cannot be scheduled before q_i for $i \geq 2$.

Assume that p_i is the first job for which this might be possible. This induces that job p_i can be scheduled between q_{i-1} and q_i , implying $p_i = 2 \cdot p_{i-1} \leq x_{i-1} + p_{i-1} + y_i$, or equivalently, $p_{i-1} \leq x_{i-1} + y_i$. In this case we present the online algorithm ON , instead of a job with length $p_i = 2 \cdot p_{i-1}$, an alternative job \hat{p}_i with length $\hat{p}_i = x_{i-1} + p_{i-1} + y_i + \epsilon$, see Figure 3. Clearly, this alternative job can only be scheduled after q_i . We show that by scheduling this alternative job, the online algorithm ON does not have a competitive ratio strictly less than 2.

First, we derive the following three claims from the assumption that ON has competitive ratio strictly less than 2:

- **Claim 1:** $q_j \leq \frac{1}{2}(y_j + x_{j-1} + p_{j-1})$, $\forall j < i$
- **Claim 2:** Either $q_i \leq \sum_{j=1}^{i-1} q_j$ or $\sum_{i=1}^{j-2} (y_i + x_i) + x_0 < 1$
- **Claim 3:** $\sum_{j=1}^{i-1} (y_j + x_{j-1} + p_{j-1}) < \sum_{j=1}^i q_j$

Claim 1: $q_j \leq \frac{1}{2}(y_j + x_{j-1} + p_{j-1})$, $\forall j < i$

Proof: Due to the choices of i , we have

$$p_j > x_j + y_{j+1}, \quad \forall 1 \leq j \leq i-2, \quad (1)$$

The claim is now proven by induction using (1). For $j = 1$ the claim holds, since $q_1 = x_0 < p_0$. Assume, the claim holds for $j-1$.

If $q_j = y_{j-1}$, we get

$$q_j = y_{j-1} \leq y_{j-1} + x_{j-2} \stackrel{(1)}{<} p_{j-2} \leq \frac{1}{2}p_{j-1} \leq \frac{1}{2}(y_j + x_{j-1} + p_{j-1})$$

In the second case, where $q_j = q_{j-1}$, we get

$$q_j = q_{j-1} \leq \frac{1}{2}(x_{j-2} + p_{j-2} + y_{j-1}) \stackrel{(1)}{<} \frac{1}{2}p_{j-1} \leq \frac{1}{2}(y_j + x_{j-1} + p_{j-1})$$

Finally, if $q_j = x_{j-1}$, we get

$$q_j = x_{j-1} = \frac{1}{2}(x_{j-1} + x_{j-1}) <_{(1)} \frac{1}{2}(x_{j-1} + p_{j-1} - y_j) \leq \frac{1}{2}(y_j + x_{j-1} + p_{j-1})$$

In all three cases, the claim also holds for j . By induction Claim 1 follows.

Claim 2: Either $q_i \leq \sum_{j=1}^{i-1} q_j$ or $\sum_{j=1}^{i-2} (y_j + x_j) + x_0 < 1$

Proof: If q_i is equal to q_{i-1} then $q_i \leq \sum_{j=1}^{i-1} q_j$. Thus, suppose that q_j equals $\max\{y_{i-1}, x_{i-1}\}$. Since ON has competitive ratio strictly less than 2 after scheduling job p_{i-1} , we have:

$$x_0 + p_0 + \sum_{j=1}^{i-1} (y_j + q_j + x_j + p_j) < 2 \cdot \left(\sum_{j=1}^{i-1} q_j + p_{i-1} \right),$$

or equivalently:

$$y_{i-1} + x_{i-1} < \sum_{j=1}^{i-1} q_j + p_{i-1} - \sum_{j=0}^{j-2} p_j - \sum_{j=1}^{i-2} (y_j + x_j) - x_0 \quad (2)$$

Furthermore, by definition of the job lengths we have:

$$p_{i-1} - \sum_{j=0}^{i-2} p_j = p_1 - p_0 = \max\{1, x_0 + y_1\} \quad (3)$$

Using (2) and (3), we get the following bound for q_i :

$$\begin{aligned} q_i &= \max\{y_{i-1}, x_{i-1}\} \leq y_{i-1} + x_{i-1} \\ &<_{(2)} \sum_{j=1}^{i-1} q_j + p_{i-1} - \sum_{j=0}^{i-2} p_j - \sum_{j=1}^{i-2} (y_j + x_j) - x_0 \\ &=_{(3)} \sum_{j=1}^{i-1} q_j + \max\{1, x_0 + y_1\} - \sum_{j=1}^{i-2} (y_j + x_j) - x_0 \end{aligned} \quad (4)$$

If $q_i > \sum_{j=1}^{i-1} q_j$, this implies $\max\{1, x_0 + y_1\} > \sum_{j=1}^{i-2} (y_j + x_j) + x_0$. This last inequality can only hold if $1 > \sum_{j=1}^{i-2} (y_j + x_j) + x_0$, proving Claim 2.

Claim 3: $\sum_{j=1}^{i-1} (y_j + x_{j-1} + p_{j-1}) < \sum_{j=1}^i q_j$

Proof: Since we assume that ON has a competitive ratio strictly less than 2, after scheduling the alternative job \hat{p}_i directly after job q_i , we have:

$$x_0 + p_0 + \sum_{j=1}^{i-1} (y_j + q_j + x_j + p_j) + y_i + q_i + \hat{p}_i < 2 \cdot \left(\sum_{j=1}^i q_j + \hat{p}_i \right).$$

Substituting $\hat{p}_i = x_{i-1} + p_{i-1} + y_i$, we get:

$$x_0 + p_0 + \sum_{j=1}^{i-2} (y_j + q_j + x_j + p_j) + y_{i-1} + q_{i-1} + q_i < 2 \cdot \sum_{j=1}^i q_j,$$

which can be rewritten as $\sum_{j=1}^{i-1} (y_j + x_{j-1} + p_{j-1}) < \sum_{j=1}^i q_j$, proving Claim 3.

In the following, we show that both possibilities in Claim 2 lead to a contradiction. Case 1, $q_i \leq \sum_{j=1}^{i-1} q_j$. Using Claim 1 and 3, we get the following contradiction:

$$\begin{aligned} \sum_{j=1}^{i-1} (y_j + x_{j-1} + p_{j-1}) &<_{(Claim3)} \sum_{j=1}^i q_j = q_i + \sum_{j=1}^{i-1} q_j \\ &\leq 2 \cdot \sum_{j=1}^{i-1} q_j \leq_{(Claim1)} 2 \cdot \sum_{j=1}^i \frac{1}{2} (y_j + x_{j-1} + p_{j-1}) \end{aligned}$$

Case 2, $q_i > \sum_{j=1}^{i-1} q_j$ and $\sum_{j=1}^{i-2} (y_j + x_j) + x_0 < 1$. Since in this case $q_i = \max\{x_{i-1}, y_{i-1}\}$, we get (4) as in the proof of Claim 2. From $\sum_{j=1}^{i-2} (y_j + x_j) + x_0 < 1$ we get $\max\{1, x_0 + y_1\} = 1$. Note that $q_{j-1} = \max\{x_0, \dots, x_{j-2}, y_1, \dots, y_{j-2}\}$. Thus,

$$\begin{aligned} \sum_{j=1}^i q_j &= q_i + \sum_{j=1}^{i-1} q_j <_{(4)} 2 \cdot \sum_{j=1}^{i-1} q_j + 1 - \sum_{j=1}^{i-2} (y_j + x_j) - x_0 \\ &< 2 \cdot \sum_{j=1}^{i-2} q_j + q_{i-1} + 1 \end{aligned} \tag{5}$$

By the definition of p_i we have:

$$\begin{aligned} \sum_{j=1}^{i-1} p_{j-1} &= p_0 + \sum_{j=1}^{i-2} p_j = p_0 + (2^{i-2} - 1) \cdot p_1 \\ &\geq 1 + 2^{i-1} - 2 = 2^{i-1} - 1 \end{aligned} \tag{6}$$

From $\sum_{j=1}^{i-2} (y_j + x_j) + x_0 < 1$ we also get $q_j \leq 1$ for all $j = 1, \dots, i-1$. Putting this together with (5), (6) and Claim 3 we have:

$$\begin{aligned}
2^{i-1} - 1 &<_{(6)} \sum_{j=1}^{i-1} p_{j-1} <_{(Claim3)} \sum_{j=1}^i q_j - \sum_{j=1}^{i-1} (y_j + x_{j-1}) \\
&<_{(5)} 2 \cdot \sum_{j=1}^{i-2} q_j + q_{i-1} + 1 - \sum_{j=1}^{i-1} (y_j + x_{j-1}) \\
&< 2 \cdot \sum_{j=1}^{i-2} q_j + 1 < 2(i-2) + 1 = 2i - 3
\end{aligned}$$

Leading to a contradiction for $i \geq 2$.

Therefore, no job p_i which can be scheduled before q_i can exist, and an online algorithm ON with competitive ratio strictly less than 2 can only schedule the jobs in the same order as they appear in σ_n . \square

Lemma 2 *If an online algorithm has a competitive ratio strictly less than 2,*

$$\frac{\sum_{i=1}^n q_i - \sum_{i=1}^n (y_i + x_i) - x_0 - p_0 + p_1}{\sum_{i=1}^n q_i + 2^{n-1} \cdot p_1} \rightarrow 0 \quad (7)$$

if $n \rightarrow \infty$.

Proof: Having a competitive ratio strictly less than 2, implies that the numerator of (7) is positive for all $n = 1, 2, 3, \dots$. Thus, after scheduling job p_j we have

$$x_j + y_j < \sum_{i=1}^j q_i - \sum_{i=1}^{j-1} (y_i + x_i) - x_0 - p_0 + p_1 \quad (8)$$

Whenever $q_{i+1} > q_i$, we have by definition of q_{i+1} 's length, that either delay y_i or x_i is larger than q_i . In equation (8), all such q_{i+1} cancel out against either y_i or x_i . So,

$$\begin{aligned}
x_j + y_j &< \sum_{i=1}^{j-1} (q_{i+1} - y_i - x_i) + q_1 - x_0 - p_0 + p_1 \\
&\leq \sum_{\substack{i=1 \\ q_{i+1}=q_i}}^{j-1} (q_{i+1} - y_i - x_i) - p_0 + p_1 \\
&\leq \sum_{\substack{i=1 \\ q_{i+1}=q_i}}^{j-1} q_{i+1} - p_0 + p_1 \quad (9)
\end{aligned}$$

Suppose now, that $q_{j+1} > q_j$. By definition, $q_{j+1} = \max\{x_j, y_j\}$. So, by in-

equality (9) we can bound q_{j+1} 's length,

$$q_{j+1} \leq \sum_{\substack{i=2 \\ q_i=q_{i-1}}}^j q_i - p_0 + p_1$$

Along the same line of argumentation we can bound the numerator of (7).

$$\begin{aligned} & \sum_{i=1}^n q_i - \sum_{i=1}^n (y_i + x_i) - x_0 - p_0 + p_1 \leq \\ & \sum_{\substack{i=1 \\ q_{i+1}=q_i}}^{n-1} (q_{i+1} - y_i - x_i) - y_n - x_n - p_0 + p_1 \leq \sum_{\substack{i=2 \\ q_i=q_{i-1}}}^n q_i - p_0 + p_1 \end{aligned}$$

Consider now the case, $q_j < q_{j+1} = \dots = q_{j+k}$. Given the above, we can bound the numerator of (7) after scheduling job p_{j+k} as follows

$$\begin{aligned} \sum_{i=1}^{j+k} q_i - \sum_{i=1}^{j+k} (y_i + x_i) - x_0 - p_0 + p_1 & \leq \sum_{\substack{i=2 \\ q_i=q_{i-1}}}^{j+k} q_i - p_0 + p_1 \\ & = \sum_{\substack{i=2 \\ q_i=q_{i-1}}}^j q_i + \sum_{i=j+2}^{j+k} q_i - p_0 + p_1 \\ & = \sum_{\substack{i=2 \\ q_i=q_{i-1}}}^j q_i - p_0 + p_1 + (k-1) \cdot q_{j+1} \\ & \leq k \cdot \left(\sum_{\substack{i=2 \\ q_i=q_{i-1}}}^j q_i - p_0 + p_1 \right) \end{aligned} \quad (10)$$

The bound on the numerator of (7) grows with a factor k , whenever the q -value doesn't change for k steps. Strongest growth is attained if $k = 2$. This means that the numerator of (7) is bounded by $O(2^{\frac{n}{2}})$, since an online algorithm can at most double the length of the q -jobs every other step. This proves the lemma. \square

4 Parallel Jobs on m Machines

In the previous sections we have given job sequences which result in a tight lower bound of 2 for the competitive ratio in the two machine case. In this section we extend this construction to the m -machine case. Besides some concrete lower bounds, we also show that by constructing job sequences similar

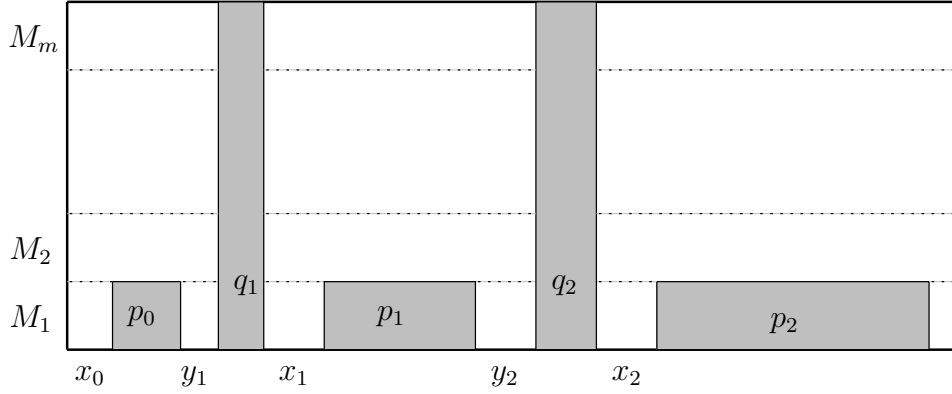


Fig. 4. Structure of the online schedule with m machines

to the ones used by Johannes [4] and in the previous sections, no lower bound greater than 2.5 can be obtained. Since the currently best upper bound on the competitive ratio is 8 (see Ye and Zhang [6]), the gap between the lower and upper bound for the m machine case, can only be closed by either considering complete different job sequences to yield better lower bounds or by developing much better online algorithms.

We define σ_{m-1} as the sequence of jobs $(p_0, q_1, p_1, q_2, p_2, \dots, q_{m-1}, p_{m-1})$, where p_i (q_i) denotes a job with processing time p_i (q_i) and a machine requirement of 1 (m). The job lengths of p_0 and all jobs q_i are as in Section 2. For jobs p_i we have

$$p_i = x_{i-1} + p_{i-1} + y_i + \epsilon \quad \forall 2 \leq i \leq m-1. \quad (11)$$

Again x_i and y_i are values given by delays the online algorithm has used for placing jobs p_i and q_i respectively. By definition of the job lengths the jobs can only be scheduled in the order of the sequence σ_{m-1} . As a consequence, Figure 4 illustrates the structure of the online schedule. An optimal schedule for σ_{m-1} is obtained by scheduling the jobs p_0, \dots, p_{m-1} parallel to each other on the m different machines, after a block containing the jobs q_1, \dots, q_n . To simplify notation for the remaining, we again let ϵ go to zero and omit it from the rest of the analysis.

If an online algorithm is ρ competitive for σ_{m-1} , the following linear inequalities constraints have to be fulfilled.

$$x_0 + p_0 \leq \rho \cdot p_0 \quad (12)$$

$$x_0 + p_0 + \sum_{j=1}^i (y_j + q_j + x_j + p_j) \leq \rho \cdot \left(\sum_{j=1}^i q_j + p_i \right) \quad \forall 1 \leq i \leq m-1 \quad (13)$$

$$\sum_{j=1}^i (y_j + q_j + x_{j-1} + p_{j-1}) \leq \rho \cdot \left(\sum_{j=1}^i q_j + p_{i-1} \right) \quad \forall 1 \leq i \leq m-1 \quad (14)$$

Inequalities (12) and (13) state that the online solution is within a factor of ρ of the optimal, after scheduling job p_i . Inequality (14) states the same after scheduling job q_i . This construction is some how similar to the construction of Johannes [4]. The main difference is that in [4] only integer delays and processing times are considered, leading to a different definition of the processing times p_i and q_i , i.e. the additive term $+\epsilon$ is replaced by $+1$. As a consequence, the lower bounds derived in [4] are not valid lower bound for the general case of arbitrary processing times.

To derive an ILP formulation in order to check whether a given value for ρ is a lower bound on the competitive ratio based on the job sequence σ_{m-1} , we have to add to (12)-(14) constraints quaranteeing that the processing time p_i and q_i are chosen properly. Constraints (15)-(17) model the job lengths of the p -jobs and q_1 . To model the lengths of the q -jobs we employ a parameter M and a set of binary variables λ_i^y , λ_i^q and λ_i^x , where $\lambda_i^y = 0$ implies that $q_i = y_{i-1}$, $\lambda_i^q = 0$ that $q_i = q_{i-1}$ and $\lambda_i^x = 0$ that $q_i = x_{i-1}$. Constraints (18)-(20) guarantee that $q_i \geq \max\{y_{i-1}, q_{i-1}, x_{i-1}\}$ holds. Constraint (21) states that exactly one of λ_i^y , λ_i^q and λ_i^x equals 0 for all i . Together with constraints (22)-(24) the equation $q_i = \max\{y_{i-1}, q_{i-1}, x_{i-1}\}$ is guaranteed. Note that M should be large enough.

$$p_0 = 1 \quad (15)$$

$$p_i = x_{i-1} + p_{i-1} + y_i \quad \forall 1 \leq i \leq m-1 \quad (16)$$

$$q_1 = x_0 \quad (17)$$

$$y_{i-1} \leq q_i \quad \forall 2 \leq i \leq m-1 \quad (18)$$

$$q_{i-1} \leq q_i \quad \forall 2 \leq i \leq m-1 \quad (19)$$

$$x_{i-1} \leq q_i \quad \forall 2 \leq i \leq m-1 \quad (20)$$

$$\lambda_i^y + \lambda_i^q + \lambda_i^x = 2 \quad \forall 2 \leq i \leq m-1 \quad (21)$$

$$q_i \leq y_{i-1} + M \cdot \lambda_i^y \quad \forall 2 \leq i \leq m-1 \quad (22)$$

$$q_i \leq q_{i-1} + M \cdot \lambda_i^q \quad \forall 2 \leq i \leq m-1 \quad (23)$$

$$q_i \leq x_{i-1} + M \cdot \lambda_i^x \quad \forall 2 \leq i \leq m-1 \quad (24)$$

The variables y_i, q_i, x_i, p_i are nonnegative and λ_i^y, λ_i^q and λ_i^x are binary variables.

Lemma 3 *If there exists no solution satisfying constraints (12)-(24), ρ is a lower bound on the competitive ratio of any online algorithm for $Pm|online - list, m_j|C_{max}$.*

Proof: Suppose there exists an ρ -competitive online algorithm. This algorithm will yield values of x_i and y_i such that constraints (12)-(24) are satisfied. \square

# machines	2	3	4	5	6	7	8	9	10	20	30
LB	1.707	1.999	2.119	2.201	2.254	2.295	2.323	2.340	2.354	2.413	2.43

Table 1

Lower Bounds on the Competitive Ratio

Based on Lemma 3, we obtain a lower bound on the competitive ratio by checking infeasibility of the constraint set (12)-(24) for a given ρ and m . To find the best possible lower bound, we employ binary search on ρ . The results of this search are displayed in Table 1.

Since σ_{m-1} contains exactly m jobs with a machine requirement of 1, these jobs can be scheduled parallel to each other on the m different machines in the offline solution. Let σ_n be a job sequence defined the same as σ_{m-1} , but now with $n \geq m$. With more than m p -jobs, one might expect a more efficient packing in the optimal offline solution. The ILP formulation for such a longer sequences become much more involved while the lower bound increases only slightly. The following theorem explains why there is only such a slight increase.

Theorem 2 *With job sequence σ_n , no lower bound on the competitive ratio larger than 2.5 can be proven for $Pm|online - list, m_j|C_{max}$.*

Proof: Consider an online algorithm which chooses $x_i = p_i$ and $y_i = 0$ for all i . As a consequence, $p_i = 2 \cdot p_{i-1}$ and $q_i = x_{i-1} = p_{i-1}$. This results in an online schedule with makespan

$$2 \cdot \sum_{j=0}^i p_j + \sum_{j=1}^i q_j = 3 \cdot \sum_{j=0}^{i-1} p_j + 2 \cdot p_i = 5 \cdot \sum_{j=0}^{i-1} p_j + 2$$

after scheduling job p_i , and a makespan of

$$2 \cdot \sum_{j=0}^{i-1} p_j + \sum_{j=1}^i q_j = 3 \cdot \sum_{j=0}^{i-1} p_j = 6 \cdot \sum_{j=0}^{i-2} p_j + 3$$

after scheduling job q_i .

Since the p -jobs grow with a factor of 2, the makespan of the optimal offline schedule equals $\sum_{j=1}^i q_j + p_i = 2 \cdot \sum_{j=0}^{i-1} p_j + 1$ after job p_i and $\sum_{j=1}^i q_j + p_{i-1} = \sum_{j=0}^{i-1} p_j + p_{i-1} = 3 \cdot \sum_{j=0}^{i-2} p_j + 2$ after job q_i .

Both after scheduling p_i and q_i the competitive ratio is less or equal to 2.5. So, with this type of job sequence no lower bound on the competitive ratio larger than $2\frac{1}{2}$ can be proven for $Pm|online - list, m_j|C_{max}$. \square

Note that even when the length of the p -jobs is defined such that $p_i \geq x_{i-1} + p_{i-1} + y_i$ Theorem 2 holds.

5 Conclusions and Remarks

In this paper we have shown that there does not exist an online algorithm for $P2|online - list, m_j|C_{max}$ with competitive ratio strictly less than 2. Thereby, we not only improve the existing lower bound on the competitive ratio of $1 + \sqrt{\frac{2}{3}}$, but also close the gap with the upper bound.

Although, greedy is the best possible in the two machine case, it is certainly not for the case with m machines. With m machines an greedy algorithm has competitive ratio m , while the best know upper bound on the competitive ratio for an arbitrary number of machines is 8. For the case with $m > 2$ we have derived lower bounds using an ILP formulation. However, the instance construction used can not give lower bounds larger than 2.5. Thus, there is still a large gap between the lower and upper bounds for the problem with m machines. We conjecture that neither the lower bound nor the upper bound is tight. So, for future research it would be interesting to improve both the lower and the upper bounds of the competitive ratio for this problem.

References

- [1] S. Albers. Online algorithms: a survey. *Mathematical Programming*, 97:3–26, 2003.
- [2] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.
- [3] W.T. Chan, F.Y.L. Chin, D. Ye, G. Zhang, and Y. Zhang. On-line scheduling of parallel jobs on two machines. *Journal of Discrete Algorithms (to appear)*, 2007.
- [4] B. Johannes. Scheduling parallel jobs to minimize the makespan. *Journal of Scheduling*, 9:433–452, 2006.
- [5] K. Pruhs, J. Sgall, and E. Torng. Online scheduling. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, editor Joseph Y-T. Leung, 2004.
- [6] D. Ye and G. Zhang. On-line scheduling of parallel jobs. *Lecture Notes in Computer Science*, 3104:279–290, 2004.