Available online at www.sciencedirect.com
ScienceDirect
Operations Research Letters 36 (2008) 51–56
Operations Research Letters
www.elsevier.com/locate/orl

Online scheduling of parallel jobs on two machines is 2-competitive

J.L. Hurink, J.J. Paulus*

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

Received 8 February 2007; accepted 20 April 2007
Available online 30 June 2007

Abstract

We consider online scheduling of parallel jobs on parallel machines. For the problem with two machines and the objective of minimizing the makespan, we show that 2 is a tight lower bound on the competitive ratio. For the problem with m machines, we derive lower bounds using an ILP formulation.

Keywords: Online 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 [4].

In this paper we study the problem of online scheduling of parallel jobs on 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. As soon as a job gets known, it has to be scheduled irrevocably (i.e. its start time has to be set) without knowing the characteristics of future jobs. Preemption is not allowed and the objective is to minimize the makespan. Adopting the notation from [4,5], this problem is denoted by $P|online-list, m_j|C_{\max}$. In this paper we show that for the problem with two machines no online algorithm can have competitive ratio strictly less than 2. For the general problem we derive new lower bounds using an ILP formulation.

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 $\sup_\sigma C_{ON}(\sigma)/C_{OPT}(\sigma) \leqslant \rho$. For background information on online algorithms, see e.g. [1,2], and on online scheduling, see e.g. [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. On the other hand, 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. For the case where jobs arrive in non-decreasing order of processing times, Chan et al. [3] give an optimal $\frac{3}{2}$-competitive algorithm. And for the case where jobs arrive in non-increasing order of processing times, 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. To derive lower bounds on the competitive ratio of online algorithms for $Pm|online-list, m_j|C_{\max}$, an enumeration scheme was proposed in [4]. However, this scheme has the drawback that it only allows for integer processing times and integer staring times of jobs. Therefore, the derived lower bounds are

* Corresponding author. Tel.: +31 534893421; fax: +31 534893402.
E-mail address: j.j.paulus@ewi.utwente.nl (J.J. Paulus).

0167-6377/$ - see front matter
doi:10.1016/j.orl.2007.06.001

only valid for a restricted version of the problem and, as a consequence, the lower bound of $1+\sqrt{\frac{2}{3}}$ for the two machine case is till now also the best known lower bound for the m-machine case.

In Sections 2 and 3, we prove that for the problem with two machines 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 new lower bounds for $Pm|online-list, m_j|C_{\max}$ using an ILP formulation. A lower bound of 2.43 is obtained for the competitive ratio for any online algorithm for $P|online-list, m_j|C_{\max}$. 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 for two machines

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 *ON* to be an online algorithm with competitive ratio $2-\delta$, with δ a small positive value, 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 $(p_0, q_1, p_1, q_2, p_2, \ldots, q_n, p_n)$, 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

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

where x_i and y_i are values given by delays the online algorithm has used for placing earlier jobs, and ε is a small positive value. This means that the job lengths are depending on the online algorithm *ON*. The concrete definition of these values is given in the next paragraph.

We prove that any online algorithm with competitive ratio strictly less than 2 has to schedule the jobs in the same order as they appear in the sequence σ_n. As a consequence, Fig. 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.

To simplify the notation for the remaining, we let $Q_n = \sum_{i=1}^n q_i$ denote the sum of processing times of the q-jobs and let $D_n = x_0 + \sum_{i=1}^n (y_i + x_i)$ denote the total delay on the jobs. Using the fact that the jobs are scheduled in the same order as they appear in σ_n, the makespan of the online schedule for σ_n is given by

$$\begin{aligned}
C_{ON}(\sigma_n) &= x_0 + p_0 + \sum_{i=1}^{n}(y_i + q_i + x_i + p_i)\\
&= \sum_{i=0}^{n} p_i + Q_n + D_n.
\end{aligned}$$

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

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

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

$$\begin{aligned}
\frac{C_{ON}(\sigma_n)}{C_{OPT}(\sigma_n)} &= \frac{\sum_{i=0}^n p_i + Q_n + D_n}{Q_n + p_n}\\
&= \frac{p_0 + (2^n - 1) \cdot p_1 + Q_n + D_n}{Q_n + 2^{n-1} \cdot p_1}\\
&= 2 - \frac{Q_n - D_n - p_0 + p_1}{Q_n + 2^{n-1} \cdot p_1}.
\end{aligned}$$

In Lemma 1 we prove that for an online algorithm *ON* with competitive ratio $2-\delta$ we have $Q_i + q_{i+1} < p_i$ and $x_i + y_{i+1} < p_i$. This last inequality implies that the online algorithm *ON* schedules the jobs in the order as they appear in σ_n. This can be seen as follows. By definition of the length of job q_i there is no gap in the schedule before p_{i-1} in which job q_i can be scheduled. The same holds for p_1. When considering job p_i, the largest gap for a job with machine requirement of 1 has size $x_{i-1} + y_i + p_{i-1}$. Due to the inequality $x_i + y_{i+1} < p_i$, this gap is smaller than $2 \cdot p_{i-1} = p_i$. Thus, p_i can only be scheduled after q_i.

In Lemma 2 we prove that for an online algorithm *ON* with competitive ratio $2-\delta$

$$\frac{Q_n - D_n - p_0 + p_1}{Q_n + 2^{n-1} \cdot p_1} \to 0$$

as n goes to infinity. 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.

To complete the proof, in the following section the proof of the two lemmata are given.

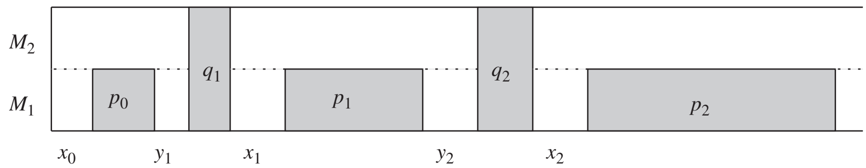

Fig. 1. Structure of the online schedule for σ_2.

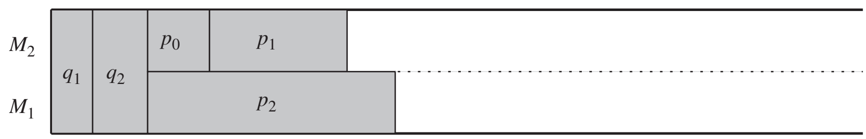

Fig. 2. Structure of the optimal offline schedule for σ_2.

3. Proof of the lemmata

Lemma 1. *If an online algorithm ON has a competitive ratio of* $2-\delta$, *then*

$$Q_i + q_{i+1} < p_i \tag{1}$$

and

$$x_i + y_{i+1} < p_i. \tag{2}$$

Proof. We prove (1) and (2) simultaneously by induction on i. If ε is chosen sufficiently small then the following inequalities follow from the $(2-\delta)$-competitiveness of algorithm *ON*:

- $x_0 < p_0$: After scheduling job p_0 we have $x_0+p_0 \leqslant (2-\delta)\cdot p_0$.
- $y_1 < p_0$: After scheduling job q_1 we have $x_0 + p_0 + y_1 + q_1 \leqslant (2-\delta)\cdot(q_1+p_0)$, or equivalently $x_0+y_1 \leqslant (1-\delta)\cdot(q_1+p_0)$. Using $q_1 = x_0 + \varepsilon$ and ε small enough, the inequality follows.
- $x_1 < x_0$: After scheduling job p_1 we have $x_0+p_0+y_1+q_1+x_1+p_1 \leqslant (2-\delta)\cdot(q_1+p_1)$, or equivalently $x_0+p_0+y_1+x_1 \leqslant (1-\delta)\cdot(q_1+p_1)$. Using $p_1 = x_0+p_0+y_1$, $q_1 = x_0+\varepsilon$ and ε small enough, the inequality follows.

By definition $q_2 = \max\{x_0, y_1, x_1\}$. Combining this with the above, we get $q_2 < p_0$. Thus, $q_1 + q_2 < x_0 + p_0 \leqslant p_1$ and (1) holds for $i=1$.

To prove that (1) holds for $i \geqslant 2$ we assume that both (1) and (2) hold up to $i-1$. Since (2) holds up to $i-1$, the jobs up to job q_{i+1} are scheduled in the order as they are in σ_n up to q_{i+1}. Since *ON* is $2-\delta$-competitive, after scheduling job p_i we have

$$\frac{p_0 + (2^i-1)\cdot p_1 + Q_i + D_i}{Q_i + 2^{i-1}\cdot p_1} \leqslant 2-\delta$$

which implies that

$$D_i - x_0 - y_1 < Q_i. \tag{3}$$

(This inequality is also used in the proof of Lemma 2.) By definition of the length of q_{i+1} we have either $q_{i+1} = q_i \leqslant Q_i$ or $q_{i+1} = \max\{x_i, y_i\} \leqslant D_i - x_0 - y_1 < Q_i$ since $i \geqslant 2$. Combining this with the induction hypothesis we have

$$Q_i + q_{i+1} < 2\cdot Q_i < 2\cdot p_{i-1} = p_i$$

and (1) also holds for i.

To prove that (2) holds for i we assume that (1) holds up to i. Since *ON* is $2-\delta$-competitive after scheduling job q_{i+1} we have

$$\frac{p_0 + (2^i-1)\cdot p_1 + Q_i + q_{i+1} + D_i + y_{i+1}}{Q_i + q_{i+1} + 2^{i-1}\cdot p_1} \leqslant 2-\delta$$

which implies that

$$D_i + y_{i+1} - x_0 - y_1 < Q_i + q_{i+1}. \tag{4}$$

(This inequality is also used in the proof of Lemma 2.) Combining this with (1), we have

$$x_i + y_{i+1} \leqslant D_i + y_{i+1} - x_0 - y_1 < Q_i + q_{i+1} < p_i$$

and (2) also holds for i. $\square$

Lemma 2. *If an online algorithm ON has a competitive ratio of* $2-\delta$, *then*

$$\frac{Q_n - D_n - p_0 + p_1}{Q_n + 2^{n-1}\cdot p_1} \to 0 \tag{5}$$

if $n \to \infty$.

Proof. We prove that (5) holds by bounding the asymptotic growth of Q_n, i.e. by showing that $Q_n \in \mathrm{O}(1.8^n)$. Since the denominator of (5) is in $\Omega(2^n)$, this proves the lemma.

We claim that either $Q_{i+1} \leqslant 1.8\cdot Q_i$ or $Q_{i+2} \leqslant 3.2\cdot Q_i$. Combining this with the fact that Q_n is monotone and $1.8^2 > 3.2$, we have that $Q_n \in \mathrm{O}(1.8^n)$.

To prove the claim, assume that $Q_{i+1} > 1.8\cdot Q_i$. This implies that $q_{i+1} > 0.8\cdot Q_i$ by definition of Q_{i+1}, and that $D_i - x_0 - y_1 > 0.8\cdot Q_i$ since the value of q_{i+1} is attained by one of the delays.

Now consider q_{i+2}. If $q_{i+2} > q_{i+1}$, then by using (3)

$$q_{i+2} \leqslant y_{i+1} + x_{i+1} = D_{i+1} - D_i \leqslant Q_{i+1} - 0.8\cdot Q_i$$

and by using (4)

$$Q_{i+2} = Q_{i+1} + q_{i+2} \leqslant 2 \cdot Q_{i+1} - 0.8 \cdot Q_i \leqslant (4 - 0.8) \cdot Q_i.$$

On the other hand, if $q_{i+2} = q_{i+1}$, we get $Q_{i+2} = Q_i + 2 \cdot q_{i+1} \leqslant 3 \cdot Q_i$, since $q_{i+1} \leqslant Q_i$.

So in both cases the claim is true, and we have proven the lemma. □

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 to the ones used in [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 [6]) and the best lower bound is 2 (from the previous section), the gap between the lower and upper bound for the m-machine case can only be closed by either considering completely 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 $(p_0, q_1, p_1, q_2, p_2, \ldots, q_{m-1}, p_{m-1})$, 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, p_1 and all jobs q_i are as in Section 2. For jobs p_i we have

$$p_i = x_{i-1} + p_{i-1} + y_i + \varepsilon \quad \forall 2 \leqslant i \leqslant m-1.$$

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, Fig. 3 illustrates the structure of the online schedule. An optimal schedule for σ_{m-1} is obtained by scheduling the jobs $p_0, \ldots, p_{m-1}$ parallel to each other on the m different machines, after a block containing the jobs $q_1, \ldots, q_{m-1}$. To simplify notation for the remaining, we 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 have to be fulfilled:

$$x_0 + p_0 \leqslant \rho \cdot p_0, \tag{6}$$

$$x_0 + p_0 + \sum_{j=1}^{i}(y_j + q_j + x_j + p_j) \leqslant \rho \cdot \left(\sum_{j=1}^{i} q_j + p_i\right) \quad \forall 1 \leqslant i \leqslant m-1, \tag{7}$$

$$\sum_{j=1}^{i}(y_j + q_j + x_{j-1} + p_{j-1}) \leqslant \rho \cdot \left(\sum_{j=1}^{i} q_j + p_{i-1}\right) \quad \forall 1 \leqslant i \leqslant m-1. \tag{8}$$

Inequalities (6) and (7) state that the online solution is within a factor of ρ of the optimal, after scheduling job p_i. Inequality (8) states the same after scheduling job q_i. This construction is somehow similar to the construction in [4]. The main difference is that in [4] only integer delays, processing times and starting times are considered, leading to a different definition of the processing times p_i and q_i, i.e. the additive term $+\varepsilon$ is replaced by $+1$. As a consequence, the lower bound derived in [4] (a bound of 2.25) is not a 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 (6)–(8) constraints guaranteeing that the processing time p_i and q_i are chosen properly. Constraints (9)–(11) 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 (12)–(14) guarantee that $q_i \geqslant \max\{y_{i-1}, q_{i-1}, x_{i-1}\}$ holds. Constraint (15) states that exactly one of λ_i^y, λ_i^q and λ_i^x equals 0 for all i. Together with constraints (16)–(18) the equation $q_i = \max\{y_{i-1}, q_{i-1}, x_{i-1}\}$ is guaranteed. Note that M should be large enough:

$$p_0 = 1, \tag{9}$$

$$p_i = x_{i-1} + p_{i-1} + y_i \quad \forall 1 \leqslant i \leqslant m-1, \tag{10}$$

$$q_1 = x_0, \tag{11}$$

$$y_{i-1} \leqslant q_i \quad \forall 2 \leqslant i \leqslant m-1, \tag{12}$$

$$q_{i-1} \leqslant q_i \quad \forall 2 \leqslant i \leqslant m-1, \tag{13}$$

$$x_{i-1} \leqslant q_i \quad \forall 2 \leqslant i \leqslant m-1, \tag{14}$$

$$\lambda_i^y + \lambda_i^q + \lambda_i^x = 2 \quad \forall 2 \leqslant i \leqslant m-1, \tag{15}$$

$$q_i \leqslant y_{i-1} + M \cdot \lambda_i^y \quad \forall 2 \leqslant i \leqslant m-1, \tag{16}$$

$$q_i \leqslant q_{i-1} + M \cdot \lambda_i^q \quad \forall 2 \leqslant i \leqslant m-1, \tag{17}$$

$$q_i \leqslant x_{i-1} + M \cdot \lambda_i^x \quad \forall 2 \leqslant i \leqslant m-1. \tag{18}$$

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 for a given m there exists no solution satisfying constraints (6)–(18), 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 a ρ-competitive online algorithm. This algorithm will yield for the job sequence σ_{m-1} values of x_i and y_i such that constraints (6)–(18) are satisfied. □

Based on Lemma 3, we obtain new lower bound on the competitive ratio by checking infeasibility of the constraint set (6)–(18) for a given ρ and m. Given an m and ρ, we check

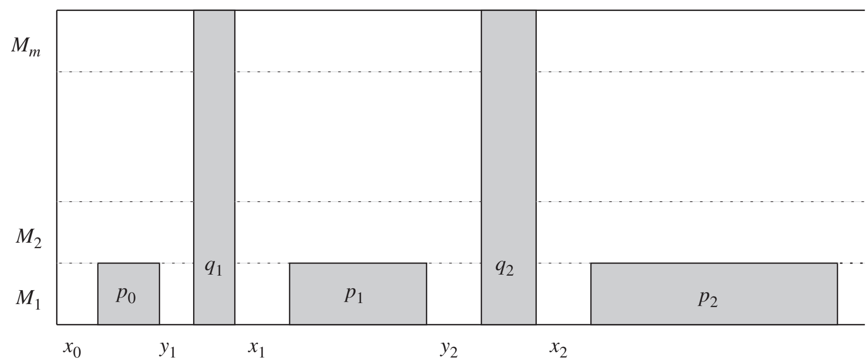

Fig. 3. Structure of the online schedule with m machines.

Table 1
Lower bounds on the competitive ratio

# 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

with an ILP solver (e.g. CPLEX) whether ρ is a lower bound by trying to find a feasible setting of the x_i's and y_i's with respect to (6)–(18). Employing binary search on ρ we get the new lower bounds displayed in Table 1. Note that a lower bound obtained for the m-machine case is also a lower bound for the $m+1$-machine case. As a result, the following theorem holds.

Theorem 2. *No online algorithm for* $P|online-list, m_j|C_{\max}$ *can have competitive ratio less than* 2.43.

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 as the same as σ_{m-1}, but now with $n \geqslant m$. With more than m p-jobs, one might expect a more efficient packing in the optimal offline solution. The ILP formulation for such longer sequences becomes much more involved while the lower bound increases only slightly. The following theorem explains why there is only such a slight increase.

Theorem 3. *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 than or equal to 2.5. So, with this type of job sequence no lower bound on the competitive ratio larger than 2.5 can be proven for $Pm|online-list, m_j|C_{\max}$. □

Note that even when the length of the p-jobs is defined such that $p_i \geqslant x_{i-1} + p_{i-1} + y_i$, Theorem 3 holds.

5. Concluding remarks

Although greedy is the best possible in the two machine case, it is certainly not for the case with m machines. With m machines a greedy algorithm has competitive ratio m, while the best known upper bound on the competitive ratio for an arbitrary number of machines is 8, see [6]. For the case with $m > 2$ we have derived lower bounds using an ILP formulation. However, the instance construction used cannot 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.

Acknowledgment

The authors are grateful to the anonymous referee for the helpful comments on an earlier draft of the paper.

References

[1] S. Albers, Online algorithms: a survey, Math. Programming 97 (2003) 3–26.
[2] A. Borodin, R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, Cambridge, 1998.
[3] W.-T. Chan, F.Y.L. Chin, D. Ye, G. Zhang, Y. Zhang, On-line scheduling of parallel jobs on two machines, J. Discrete Algorithms 2006, in press, doi: 10.1016/j.jda.2006.07.005.
[4] B. Johannes, Scheduling parallel jobs to minimize the makespan, J. Scheduling 9 (2006) 433–452.
[5] K. Pruhs, J. Sgall, E. Torng, Online scheduling, in: J.Y.-T. Leung (Ed.), Handbook of Scheduling: Algorithms, Models, and Performance Analysis, 2004.
[6] D. Ye, G. Zhang, On-line scheduling of parallel jobs, in: Lecture Notes in Computer Science (SIROCCO 2004), vol. 3104, 2004, pp. 279–290.