Optimal Non-Preemptive Semi-Online Scheduling on Two Related Machines

Leah Epstein Lene M. Favrholdt

Affiliations:

- Leah Epstein, School of Computer Science, The Interdisciplinary Center, Herzliya, Israel. `lea@idc.ac.il`
- Lene Monrad Favrholdt, Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark. `lenem@imada.sdu.dk`

Proposed Running Head:
Non-Preemptive Semi-Online Scheduling

All correspondence should be sent to:

Leah Epstein
School of Computer Science
The Interdisciplinary Center
P.O.B 167
46150 Herzliya
Israel
Email: lea@idc.ac.il

Abstract. We consider the following non-preemptive semi-online scheduling problem. Jobs with non-increasing sizes arrive one by one to be scheduled on two uniformly related machines, with the goal of minimizing the makespan. We analyze both the optimal overall competitive ratio, and the optimal competitive ratio as a function of the speed ratio ($q \geq 1$) between the two machines. We show that the greedy algorithm LPT has optimal competitive ratio $\frac{1}{4}(1 + \sqrt{17}) \approx 1.28$ overall, but does not have optimal competitive ratio for every value of q. We determine the intervals of q where LPT is an algorithm of optimal competitive ratio, and design different algorithms of optimal competitive ratio for the intervals where it fails to be the best algorithm. As a result, we give a tight analysis of the competitive ratio for every speed ratio.

Keywords: scheduling, makespan, semi-online, competitive ratio, LPT.

1 Introduction

The Problem. In this paper we study non-preemptive semi-online scheduling on two uniformly related machines. In the model of uniformly related machines, each machine has a *speed* and each job has a *size* which is the time it takes to complete it on a machine with unit speed. The jobs arrive one by one in order of non-increasing sizes. Each job must be assigned to one of the machines without any knowledge of future jobs (except for a bound on their size that follows from the size of the current job). Since the jobs are known to have non-increasing sizes, the problem cannot be seen as on-line but semi-online. We study the non-preemptive case, where it is not allowed to split a job in more parts and run the various parts on different machines. The goal is to minimize the *makespan*, i.e., the latest completion time of any job.

The processing time of a job on a given machine is also called the *load* of the job on that machine. The load of a machine is the sum of the loads of the jobs assigned to it. Thus, the makespan is the maximum load of any machine.

Since we study the case of two machines, the important parameter is the speed ratio $q \geq 1$ between the two machines. Without loss of generality, we assume that the faster machine has speed 1, and the other machine has speed $\frac{1}{q}$. We denote the faster machine by M_1 and the other machine by M_q.

Preliminaries. The quality of a semi-online algorithm, similarly to on-line algorithms, is measured by the *competitive ratio* which is the worst case ratio of the cost (the makespan, in this paper) of the semi-online algorithm to the cost of an optimal off-line algorithm which knows the whole sequence in advance.

The semi-online algorithm under consideration as well as its makespan is denoted by SONL. Similarly, the optimal off-line algorithm as well as its makespan is denoted by OPT. Thus, the competitive ratio of an algorithm SONL is

$$C = \inf\{c \mid \text{SONL} \leq c \cdot \text{OPT}, \text{ for any input sequence}\}.$$

For any $c \geq C$, SONL is said to be *c-competitive.*

The greedy algorithm LPT (Longest Processing Time first) was originally designed by Graham [6] for off-line scheduling on identical machines. It sorts the jobs by non-increasing sizes and schedules them one by one on the least loaded machine. This algorithm also works for the semi-online version where the jobs arrive in order of non-increasing sizes. The natural extension for uniformly related machines is as follows:

Algorithm LPT: Assign each arriving job J (of size p) to the machine that would finish it first. Formally, for each machine i let L_i be its load before the arrival of J. The job J is assigned to the fastest machine i for which $L_i + \frac{p}{s_i}$ is minimized.

Previous Work. All previous study of this problem on non-identical machines involves a study of the LPT algorithm. For two machines, Mireault, Orlin and Vohra [7] give a complete analysis of LPT as a function of the speed ratio. They show that the interval $q \geq 1$ is partitioned into nine intervals, and introduce a function which gives the competitive ratio in each interval (they consider the off-line problem, so they do not use the term competitive ratio). Some properties of LPT were already shown earlier. Graham [6] shows that the exact approximation ratio of LPT is $\frac{7}{6}$ for two identical machines. Seiden, Sgall and Woeginger [8] show that this is tight, i.e., LPT has the best possible competitive ratio for the problem. For two related machines, [5] shows that for any speed ratio, the performance ratio of LPT is at most $\frac{1}{4}(1+\sqrt{17}) \approx 1.28$.

For m identical machines, Graham [6] shows that the exact approximation ratio of LPT is $\frac{4}{3} - \frac{1}{3m}$. For three machines, [8] gives a general lower bound of $\frac{1}{6}(1+\sqrt{37}) \approx 1.18$. For a general setting of m related machines, Friesen [4] shows that the overall approximation ratio of LPT is between 1.52 and $\frac{5}{3}$. Dobson [2] claims to improve the upper bound to $\frac{19}{12} \approx 1.58$. Unfortunately, his proof does not seem to be complete.

Our Results. In this paper we give the exact competitive ratio as a function of the speed ratio q for semi-online non-preemptive scheduling on two related machines with non-increasing job sizes (see Figure 1). The function involves 15 distinct intervals, as defined in Theorem 1.

In some of those intervals, we give general lower bounds which match the upper bounds in [7]. In those cases, LPT is an optimal semi-online algorithm. In the other intervals, we design new algorithms and prove matching general lower bounds. We show that, in terms of overall competitive ratio, $\frac{1}{4}(1+\sqrt{17})$ is the optimal competitive ratio achieved at $q = \frac{1}{4}(1+\sqrt{17})$ by LPT.

It is interesting to examine our results in the view of the results for on-line scheduling on two related machines. Unlike our problem, for that problem, LPT is optimal for all $q \geq 1$. The overall competitive ratio is ϕ ($\phi \approx 1.618$ is the golden ratio). For $q \leq \phi$ the competitive ratio is $1+\frac{q}{q+1}$ and the competitive ratio for $q \geq \phi$ is $1+\frac{1}{q}$. There are only two distinct intervals and the worst competitive ratio is achieved at ϕ. Surprisingly, for both problems, the highest competitive ratio is equal to the value of q for which it is achieved. The upper bounds, as well as the overall lower bound are given in [1], the other lower bounds are given in [3].

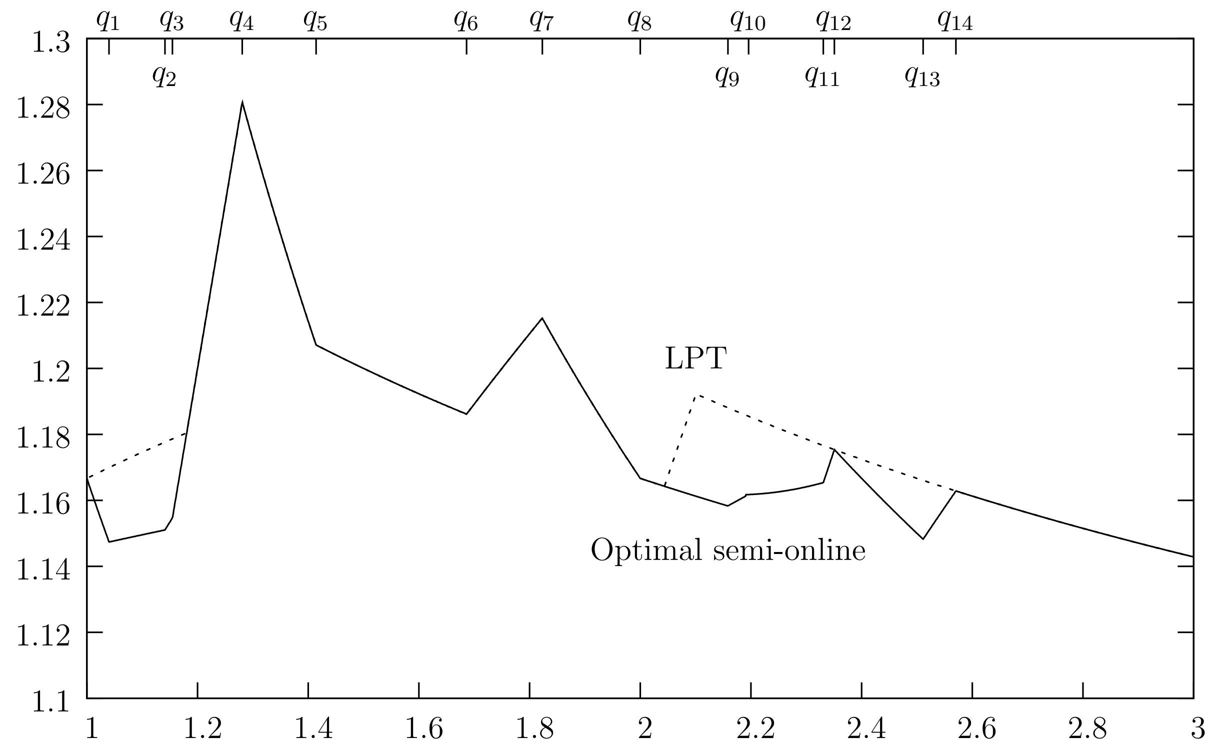

Fig. 1. The competitive ratio as a function of q

2 The Function

Theorem 1. *The optimal competitive ratio for semi-online scheduling on two related machines (with speed ratio $q \geq 1$) is described by the following function depicted in Figure 1.*

$$C(q) = \begin{cases} C_1(q), & 1 \leq q \leq q_1 \approx 1.0401 \\ C_2(q), & q_1 \leq q \leq q_2 \approx 1.1410 \\ C_3(q), & q_2 \leq q \leq \sqrt{\frac{4}{3}} \approx 1.1547 \\ C_4(q), & \sqrt{\frac{4}{3}} \leq q \leq \frac{1}{4}(1+\sqrt{17}) \approx 1.2808 \\ C_5(q), & \frac{1}{4}(1+\sqrt{17}) \leq q \leq \sqrt{2} \approx 1.4142 \\ C_6(q), & \sqrt{2} \leq q \leq \frac{1}{4}(1+\sqrt{33}) \approx 1.6861 \\ C_7(q), & \frac{1}{4}(1+\sqrt{33}) \leq q \leq \frac{1}{2}(1+\sqrt{7}) \approx 1.8229 \\ C_8(q), & \frac{1}{2}(1+\sqrt{7}) \leq q \leq 2 \\ C_9(q), & 2 \leq q \leq \frac{1}{2}(1+\sqrt{11}) \approx 2.1583 \end{cases}$$

$$
C(q) = \begin{cases} C_{10}(q), & \frac{1}{2}(1+\sqrt{11}) \le q \le q_{10} \approx 2.1956 \\ C_{11}(q), & q_{10} \le q \le q_{11} \approx 2.3307 \\ C_{12}(q), & q_{11} \le q \le \frac{1}{4}(3+\sqrt{41}) \approx 2.3508 \\ C_{13}(q), & \frac{1}{4}(3+\sqrt{41}) \le q \le q_{13} \approx 2.5111 \\ C_{14}(q), & q_{13} \le q \le q_{14} \approx 2.5704 \\ C_{15}(q), & q \ge q_{14}, \end{cases}
$$

$$
C_1(q) = \frac{2}{3} + \frac{1}{2q}, \qquad C_2(q) = 1 + \frac{1}{2}\left(4q^2 + 4q - 1 - \sqrt{(4q^2+4q-1)^2 - 4q^2}\right),
$$

$$
C_3(q) = \frac{6q+4}{3q+6}, \qquad C_4(q) = q, \qquad C_5(q) = \frac{1}{2} + \frac{1}{q},
$$

$$
C_6(q) = 1 + \frac{1}{2q+2}, \qquad C_7(q) = \frac{2q+1}{q+2}, \qquad C_8(q) = \frac{2}{3} + \frac{1}{q},
$$

$$
C_9(q) = 1 + \frac{1}{2q+2}, \qquad C_{10}(q) = \frac{3q+2}{2q+3}, \qquad C_{11}(q) = \frac{q^2+3+\sqrt{q^4-6q^2+24q+9}}{6q},
$$

$$
C_{12}(q) = \frac{q}{2}, \qquad C_{13}(q) = \frac{3}{4} + \frac{1}{q}, \qquad C_{14}(q) = 1 + \frac{q^2+2q-2-\sqrt{q^4+8q+4}}{2q+4},
$$

$$
C_{15}(q) = 1 + \frac{1}{2q+1},
$$

q_1 *is the largest real root of* $84q^4 - 24q^3 - 80q^2 + 6q + 9$,
q_2 *is the largest real root of* $27q^4 + 48q^3 - 54q^2 - 48q + 8$,
q_{10} *is the smallest real root of* $3q^4 - 9q^3 - 8q^2 + 21q + 18$,
q_{11} *is the largest real root of* $q^3 - 2q - 8$,
q_{13} *is the largest real root of* $20q^4 - 39q^3 - 46q^2 + 32q + 32$,
q_{14} *is the largest real root of* $4q^4 - 6q^3 - 12q^2 + q + 4$.

Note that the function $C(q)$ attains it maximum value of $\frac{1}{4}(1+\sqrt{17})$ at $q = \frac{1}{4}(1+\sqrt{17})$. This is the competitive ratio obtained by LPT [5]. Hence, in terms of overall competitive ratio, LPT is optimal (see also the proof of the lower bound of intervals 4 and 5 in section 5).

The proof of the upper bound is given in Section 4 and the proof of the lower bound in Section 5.

3 Properties and Assumptions

In this section we describe a few facts and assumptions used in the upper bound analysis.

We assume without loss of generality that OPT $= 1$. Thus, the ratio $\frac{\text{SONL}}{\text{OPT}} = \text{SONL}$. For a given input sequence, we denote the total size of jobs by P. Note that $P \leq 1 + \frac{1}{q} = \frac{q+1}{q}$ always holds, since the total size of jobs scheduled by OPT is at most 1 on M_1 and at most $\frac{1}{q}$ on M_q.

For the first k jobs of an input sequence, we denote the makespan of the semi-online algorithm by SONL_k. The jobs in a sequence are denoted $J_1, J_2, \ldots, J_\ell$ and their sizes are denoted $p_1, p_2, \ldots, p_\ell$. Thus, J_ℓ is the last job and $p_1 \geq p_2 \geq \ldots \geq p_\ell$.

In our lower bound proofs we consider only minimal worst-case sequences, i.e., w.l.o.g. we assume that the makespan is determined by the last (and hence the smallest) job of the sequence: $\text{SONL} > \text{SONL}_{\ell-1}$.

Lemma 1. *For a given input sequence and a given semi-online algorithm, assume that J_ℓ is scheduled according to the LPT rule and that $SONL > SONL_{\ell-1}$. Let $P_1^{\ell-1}$ and $P_q^{\ell-1}$ be the total size of jobs assigned to M_1 and M_q, respectively, just before the arrival of J_ℓ. Then,*

$$SONL = \min\{P_1^{\ell-1} + p_\ell,\ q(P_q^{\ell-1} + p_\ell)\}.$$

Proof. LPT schedules J_ℓ on a machine such that the resulting load on that machine is minimized. By the assumption that J_ℓ determines the makespan, the final makespan is equal to the load on the machine running J_ℓ. □

The following lemma appears in [7]. For completeness, we prove it here as well.

Lemma 2. *For a given input sequence and a given semi-online algorithm SONL, assume that the last job J_ℓ is scheduled according to the LPT rule and that $SONL > SONL_{\ell-1}$. Then*

$$SONL \leq 1 + \frac{q}{q+1} p_\ell.$$

Proof. Let $P_q^{\ell-1}$ and $P_1^{\ell-1}$ denote the total size of jobs assigned to the slow and the fast machine, respectively, just before the last job is assigned. Note that $P = P_q^{\ell-1} + P_1^{\ell-1} + p_\ell$. By Lemma 1, the final makespan is $\min\{q(P_q^{\ell-1} + p_\ell), P_1^{\ell-1} + p_\ell\}$. 1). This is bounded by the convex combination

$$\frac{1}{q+1} q(P_q^{\ell-1} + p_\ell) + \frac{q}{q+1}(P_1^{\ell-1} + p_\ell) = \frac{q}{q+1}(P_q^{\ell-1} + P_1^{\ell-1} + 2p_\ell) = \frac{q}{q+1}(P + p_\ell) \leq 1 + \frac{q}{q+1} p_\ell.$$

□

We will sometimes use Lemma 2 in the following form.

Corollary 1. *For a given input sequence and a given semi-online algorithm SONL, assume that the last job J_ℓ is scheduled according to the LPT rule and that $SONL > SONL_{\ell-1}$.*

$$\text{If } SONL > C, \text{ then } p_\ell > (C-1)\frac{q+1}{q} \quad .$$

Lemma 3. *For a given input sequence and a given semi-online algorithm, assume that J_ℓ is scheduled according to the LPT rule and that $SONL > SONL_{\ell-1}$. Let $k \in \mathbb{N}$.*

$$\text{If } SONL > 1 + \frac{1}{(k+1)(q+1)}, \text{ then OPT runs at most } k \text{ jobs on } M_q.$$
$$\text{If } SONL > 1 + \frac{q}{(k+1)(q+1)}, \text{ then OPT runs at most } k \text{ jobs on } M_1.$$

Proof. We prove the contrapositive.

If OPT schedules at least $k+1$ jobs on M_1, the smallest of these k jobs has size at most $\frac{1}{k+1}$. Thus, $p_\ell \le \frac{1}{k+1}$, and, using Lemma 2, we get SONL $\le 1 + \frac{q}{q+1}\frac{1}{k+1}$.

Similarly, if at least $k+1$ jobs are scheduled on M_q, $p_\ell \le \frac{1}{q(k+1)}$. Using Lemma 2, we get SONL $\le 1 + \frac{q}{q+1}\frac{1}{q(k+1)} = 1 + \frac{1}{(q+1)(k+1)}$. □

4 New Algorithms

In this section we present algorithms of optimal competitive ratio, for intervals where LPT is not optimal. We first mention all intervals where LPT is an optimal algorithm. In [7] the exact performance ratio of LPT is given. In all intervals where the lower bound in Section 5 matches the upper bound in [7], clearly LPT has optimal competitive ratio.

The names we use for the intervals are taken from the definition of the function. Hence, we deal with intervals 1–15. The right endpoint of interval i is called q_i. The intervals where LPT is optimal are as follows: The first interval is the point $q = 1$. For $q = 1$, it is known [6] that the competitive ratio of LPT is $\frac{7}{6}$ and that this is the best possible competitive ratio for any semi-online algorithm [8]. However, for $q = 1 + \varepsilon$, for small $\varepsilon > 0$, this paper shows that LPT is not an optimal semi-online algorithm.

The other intervals where LPT is optimal are $1.18 \approx \frac{1}{6}(1+\sqrt{37}) \le q \le q_{\mathrm{LPT}} \approx 2.04$ and $q \ge q_{14} \approx 2.57$, where q_{LPT} is the largest real root of $4q^3 - 4q^2 - 10q + 3$.

This leaves the following intervals to deal with. Intervals 1–4 (not including $q = 1$ in interval 1, and interval 4 only up to $\frac{1}{6}(1+\sqrt{37})$) and intervals 9–14 (interval 9 starting only at q_{LPT}). We design three new algorithms *Slow-LPT, Balanced-LPT and Opposite-LPT*. Slow-LPT has optimal competitive ratio in the interval $1 < q < \frac{1}{6}(1+\sqrt{37})$. Balanced-LPT has optimal competitive ratio in the intervals $q_{\mathrm{LPT}} < q \le q_{10}$ and $q_{12} \le q < q_{14}$, and Opposite-LPT has optimal competitive ratio in the interval $q_{10} \le q \le q_{12}$.

As can be seen in the next section, the most difficult sequences for the algorithms are quite short sequences (up to six jobs). For longer sequences, the last job is relatively small. Thus, as shown by Lemma 2, the algorithm benefits from the non-increasing order in this case. As may be seen from the definitions of the three new algorithms, the most difficult decision is either the decision for the second job, or the first and the third jobs.

4.1 The First Four Intervals

We design an algorithm Slow-LPT which has optimal competitive ratio in the interval $1 < q < \frac{1}{6}(1+\sqrt{37})$. Intuitively, the reason why LPT fails in this interval is that the slow machine is not much slower than the faster one. Since the fast machine does not dominate the slow machine so easily, it often makes sense to use the slow machine first, and keep the fast machine free for future jobs. The algorithm is actually optimal in the interval $1 \le q \le q_4$, giving an alternative algorithm with optimal competitive ratio in the interval $\frac{1}{6}(1+\sqrt{37}) \le q \le q_4$.

Algorithm Slow-LPT

Assign J_1 to M_q.
Assign J_2 to M_1.
If $q(p_1+p_3) \le C(q)(p_2+p_3)$, assign J_3 to M_q, and otherwise to M_1.
Assign the remaining jobs by the LPT rule.

To analyze the algorithm, first note that, for $i, j \in \{1,2,3,4\}$, $C_i(q) \le C_j(q)$ in interval j. Thus, in the first four intervals, $C(q) = \max\{C_1(q), C_2(q), C_3(q), C_4(q)\}$.

By Lemma 2, we need only consider sequences with $p_\ell > \frac{1}{4}$, since in these intervals

$$C(q) \ge C_2(q) > 1.145 \ge 1 + \frac{q}{4q+4} .$$

This actually means that OPT runs at most 3 jobs on each machine. Hence, we need only consider sequences of up to 6 jobs, all larger than $\frac{1}{4}$. In the following, we analyze sequences of each length $\ell = 1, \ldots, 6$ separately.

One job. Clearly, OPT $= p_1$ and SONL $= qp_1$. Thus, SONL $\le q = C_4(q) \le C(q)$.

Two jobs. The makespan of Slow-LPT is determined by the first job, since $qp_1 \ge qp_2 \ge p_2$.

Three jobs. Since OPT must assign at least two jobs on one machine, OPT $\ge p_2 + p_3$.

By the assumption that the last job determines the makespan, if Slow-LPT schedules J_3 on M_1, then SONL $= p_2 + p_3 \leq$ OPT. Otherwise, by the same assumption and the definition of the algorithm SONL $= q(p_1 + p_3) \leq C(q)(p_2 + p_3) \leq$ OPT.

Four jobs. If Slow-LPT runs J_3 on M_q, J_4 will be scheduled on M_1, since $p_2+p_4 \leq p_1+p_3 \leq q(p_1 + p_3)$. Thus, J_4 will not determine the makespan.

If Slow-LPT runs J_3 on M_1, SONL $= \min\{q(p_1 + p_4), p_2 + p_3 + p_4\}$. If OPT schedules J_1 and some other job on one machine, OPT $\geq p_1 + p_4$. In this case, $\frac{\text{SONL}}{\text{OPT}} \leq \frac{q(p_1+p_4)}{p_1+p_4} = q \leq C(q)$. Otherwise, OPT $\geq p_2 + p_3 + p_4 \geq$ SONL.

Five jobs. We split the proof in three cases, according to the way J_3 and J_4 are scheduled by Slow-LPT.

Case 1: Slow-LPT assigns J_3 to M_q.

In this case, J_4 is assigned to M_1. After scheduling J_4 (and before scheduling J_5), the total size of jobs on M_1 is at most the total size on M_q, since $p_1 + p_3 \geq p_2 + p_4$. Thus, SONL $\leq \frac{1}{2}(p_1 + p_2 + p_3 + p_4) + p_5 = \frac{1}{2}(P + p_5) \leq \frac{1}{2}(1 + \frac{1}{q} + p_5)$. Having five jobs in the sequence, we know that OPT has three jobs on one of the machines. One job must be of size at most $\frac{1}{3}$, since OPT $= 1$. This gives $p_5 \leq \frac{1}{3}$ and thus SONL $\leq \frac{1}{2}(1 + \frac{1}{q} + \frac{1}{3}) = \frac{2}{3} + \frac{1}{2q} = C_1(q)$.

Case 2: Slow-LPT assigns J_3 and J_4 to M_1.

By Lemma 1, SONL $= \min\{q(p_1 + p_5), p_2 + p_3 + p_4 + p_5\}$. As mentioned earlier, OPT runs at most three jobs on each machine, so it runs J_1 with at least one other job. Thus, OPT $\geq p_1 + p_5$, and SONL $\leq q(p_1 + p_5) \leq q \cdot$ OPT $\leq C(q)$.

Case 3: Slow-LPT assigns J_3 to M_1 and J_4 to M_q.

In this case, SONL $= \min\{q(p_1 + p_4 + p_5), p_2 + p_3 + p_5\}$.

If OPT runs J_1 with two more jobs, OPT $\geq p_1 + p_4 + p_5 \geq \frac{1}{q}$ SONL $\geq \frac{1}{C(q)}$ SONL.

We split the case where OPT runs J_1 with exactly one other job in three subcases, based on which job is combined with J_1.

Case 3.1: OPT runs J_4 or J_5 with J_1.

The machine not running J_1 is loaded by at least $p_2 + p_3 + p_5 \geq$ SONL.

Case 3.2: OPT runs J_3 with J_1.

In this case, $p_2 + p_4 + p_5 \leq 1$. Since $p_5 \leq p_4$, this implies $p_2 + 2p_5 \leq 1$. Thus, SONL $\leq p_2 + p_3 + p_5 \leq 2p_2 + p_5 = 2(p_2 + 2p_5) - 3p_5 \leq 2 - 3p_5$. Moreover, by Lemma 2, SONL $\leq 1 + \frac{q}{q+1}p_5$. The two upper bounds on SONL are equal when $p_5 = \frac{q+1}{4q+3}$ and hence SONL $\leq 1 + \frac{q}{4q+3}$. Solving $1 + \frac{q}{4q+3} \leq C_2(q)$, we get $4q^2 - 2q - 3 \leq 0$ and thus $q \leq \frac{1}{4}(1 + \sqrt{13}) \approx 1.15$, which holds throughout the first two intervals. In the third and fourth intervals, $1 + \frac{q}{4q+3} \leq C_3(q)$, as $9q^2 - 5q - 6 \geq 0$ for $q \geq \frac{1}{18}(5 + \sqrt{241}) \approx 1.140$.

Case 3.3: OPT runs J_2 with J_1.

Let $p_1 + p_2 = \alpha$ and $p_3 + p_4 + p_5 = \beta$. Then, by the way OPT schedules the jobs, either $\alpha \le 1$ and $\beta \le \frac{1}{q}$, or $\beta \le 1$ and $\alpha \le \frac{1}{q}$. Since J_3 was assigned to the fast machine by Slow-LPT, we know that $q(p_1 + p_3) > C(q)(p_2 + p_3) \ge C_2(q)(p_2 + p_3)$. This can also be expressed as $q(\alpha - p_2 + p_3) > C_2(q)(p_2 + p_3)$, which is equivalent to $p_2 < \frac{q\alpha}{q+C_2(q)} + \frac{q-C_2(q)}{q+C_2(q)}p_3$. Thus,

$$\begin{aligned}
\text{SONL} &< \frac{q\alpha}{q + C_2(q)} + \frac{q - C_2(q) + q + C_2(q)}{q + C_2(q)}\, p_3 + p_5 \\
&= \frac{q\alpha}{C_2(q) + q} + \frac{2q}{C_2(q) + q}\, p_3 + p_5 \\
&= \frac{q\alpha}{C_2(q) + q} + \frac{2q}{C_2(q) + q}\, (\beta - p_4 - p_5) + p_5 \\
&= \frac{(\alpha + 2\beta)q}{C_2(q) + q} - \frac{2q}{C_2(q) + q}\, p_4 + \frac{C_2(q) - q}{C_2(q) + q}\, p_5 \\
&\le \frac{1 + 2q}{C_2(q) + q} - \frac{2q}{C_2(q) + q}\, p_4 + \frac{C_2(q) - q}{C_2(q) + q}\, p_5, \quad \text{since } \alpha + 2\beta \le \frac{1}{q} + 2 \\
&\le \frac{2q + 1}{C_2(q) + q} + \frac{C_2(q) - 3q}{C_2(q) + q}\, p_5, \qquad\qquad\qquad\quad \text{since } p_4 \ge p_5
\end{aligned}$$

We show that this is at most $C_2(q)$. If $p_5 \le (C_2(q)-1)\frac{q+1}{q}$, SONL $\le C_2(q)$ by Corollary 1. If $p_5 > (C_2(q) - 1)\frac{q+1}{q}$, the term $C_2(q) - 3q$ in the upper bound on SONL is negative and we get

$$\text{SONL} < \frac{2q + 1}{C_2(q) + q} + \frac{C_2(q) - 3q}{C_2(q) + q}\, (C_2(q) - 1)\, \frac{q + 1}{q} = C_2(q)\,.$$

The equality is obtained by substituting $1 + \frac{1}{2}\left(4q^2 + 4q - 1 + \sqrt{(4q^2 + 4q - 1)^2 - 4q^2}\right)$ for $C_2(q)$.

Six jobs. Since we are only considering sequences where $p_\ell > \frac{1}{4}$, OPT must schedule exactly three jobs on each machine. Thus, OPT $\ge p_1 + p_5 + p_6$ and, since among J_1, J_2, and J_3, at least two run on one machine, OPT $\ge p_2 + p_3 + p_6$.

We split the proof in four cases according to how jobs J_3, J_4, and J_5 are scheduled by Slow-LPT.

Case 1: Slow-LPT runs J_3 on M_q.

By the definition of the algorithm, $q(p_1+p_3) \le C(q)(p_2+p_3)$. Furthermore, J_4 and J_5 are assigned to M_1, since $q(p_1+p_3) \ge p_2+p_4$. Thus, by Lemma 1, SONL $\le q(p_1+p_3+p_6) \le C(q)(p_2 + p_3) + qp_6 \le C(q)(p_2 + p_3 + p_6) \le C(q) \cdot$ OPT.

Case 2: Slow-LPT runs J_3 and J_4 on M_1.

Since SONL assigns J_2, J_3, and J_4 to M_1, SONL $\leq q(p_1+p_5+p_6) \leq q\,\mathrm{OPT} \leq C(q)\,\mathrm{OPT}$.

Case 3: Slow-LPT runs J_3 on M_1 and J_4 and J_5 on M_q.

In this case, SONL $\leq p_2 + p_3 + p_6 \leq$ OPT.

Case 4: Slow-LPT runs J_3 and J_5 on M_1 and J_4 on M_q.

In this case, SONL $\leq \min\{q(p_1 + p_4 + p_6), p_2 + p_3 + p_5 + p_6\}$.

We split the remaining part of the analysis in three cases based on how OPT combines J_1, J_5, and J_6.

Case 4.1: OPT does not run J_1 with both J_5 and J_6.

In this case, OPT $\geq p_1 + p_4 + p_6$, so SONL $\leq q \cdot \mathrm{OPT} \leq C(q) \cdot \mathrm{OPT}$.

Case 4.2: OPT runs J_1, J_5, and J_6 on M_q.

From the structure of OPT, $q(p_1 + p_5 + p_6) \leq q$ and $p_2 + p_3 + p_4 \leq 1$, so $p_4 \leq \frac{1}{3}$. Thus, SONL $\leq q(p_1 + p_4 + p_6) = q(p_1 + p_5 + p_6) + q(p_4 - p_5) \leq 1 + q(\frac{1}{3} - p_5) < 1 + \frac{q}{12}$, since we consider only sequences with $p_\ell > \frac{1}{4}$.

$$\begin{aligned} &\text{For } q \geq \frac{12}{11} \approx 1.09, && 1 + \frac{q}{12} \leq q \leq C(q), && \text{and} \\ &\text{for } q \leq \sqrt{10} - 2 \approx 1.16, && 1 + \frac{q}{12} \leq \frac{2}{3} + \frac{1}{2q} = C_1(q). && \end{aligned}$$

Case 4.3: OPT runs J_1, J_5, and J_6 on M_1.

In this case, SONL $\leq \min\{q(1+p_4-p_5), \frac{1}{q} - p_4 + p_5 + p_6\}$ and $p_4 \leq \frac{1}{3q}$. Let $\alpha = p_4 - p_5$. Then, $p_6 \leq p_5 \leq -\alpha + p_4 \leq -\alpha + \frac{1}{3q}$. We get SONL $\leq \min\{q(1 + \alpha), \frac{1}{q} - 2\alpha + \frac{1}{3q}\} = \min\{q(1 + \alpha), \frac{4}{3q} - 2\alpha\}$. The two terms are equal for $\alpha = \alpha_1 = \frac{4-3q^2}{q(3q+6)}$. For $q \leq \sqrt{4/3}$, α_1 is positive, and SONL $\leq q(1 + \alpha_1) = \frac{6q+4}{3q+6} = C_3(q) \leq C(q)$. For $q \geq \sqrt{4/3}$, SONL $\leq \frac{4}{3q} - 2\alpha \leq q \leq C(q)$.

4.2 Intervals 9 and 10, 13 and 14

In intervals 9–14, only sequences of five jobs and less can be slightly problematic, unlike the intervals 1–4, where sequences of six jobs had to be considered.

Both algorithms for intervals 9–14 have a special rule for the second job only. In this section, we consider the algorithm Balanced-LPT, which tries to assign the second job to M_q.

Algorithm Balanced-LPT

Assign J_1 to M_1.

If $qp_2 > C(q)(p_1 + p_2)$, assign J_2 to M_1, and otherwise to M_q.

Assign the remaining jobs by the LPT rule.

Intervals 9 and 10. Recall that $C_9(q) = 1 + \frac{1}{2q+2}$ and $C_{10}(q) = \frac{3q+2}{2q+3}$, and note that, in intervals 9 and 10, $C(q) = \max\{C_9(q), C_{10}(q)\}$.

By Lemma 3, for any sequence such that SONL $> C(q)$, OPT runs at most one job on M_q, since $C(q) \geq C_9(q) = 1 + \frac{1}{2(q+1)}$. Similarly, OPT runs at most 4 jobs on M_1, since $\frac{1}{2(q+1)} \geq \frac{q}{5(q+1)}$ for $q \leq 2.5$. This means that we need only consider sequences of at most five jobs. If an optimal off-line algorithm does not run any jobs on M_q, Balanced-LPT will not break the ratio, so we will only consider sequences where OPT schedules exactly one job on M_q and at most four on M_1.

In intervals 9 and 10, Balanced-LPT always assigns J_2 to M_q, since $C(q)(p_1 + p_2) \geq C_9(q)(p_1 + p_2) \geq 2C_9(q)p_2 \geq qp_2$ for $q \leq \frac{1+\sqrt{13}}{2} \approx 2.30$. This shows that sequences of at most two jobs cannot break the ratio.

By Lemma 1, if there are at least three jobs, SONL $\leq P - p_2$. If OPT does not run J_1 on M_q, we get OPT $\geq P - p_2$. This leaves only the case where OPT runs J_1 on M_q and all other jobs on M_1 to consider.

Three jobs. Since we consider sequences where OPT runs J_1 on M_q and all other jobs on M_1, OPT $= \max\{qp_1, p_2 + p_3\}$. By Lemma 1, SONL $\leq p_1 + p_3 \leq 2p_1 \leq qp_1 \leq$ OPT, since Balanced-LPT runs J_2 on M_q.

Four jobs. Since OPT runs J_1 on M_q and J_2, J_3, and J_4 on M_1, $p_1 \leq \frac{1}{q}$ and $p_2+p_3+p_4 \leq 1$. Combining the latter inequality with $p_2 \geq p_3 \geq p_4$ gives $p_3+p_4 \leq \frac{2}{3}$. Thus, using Lemma 1, we have SONL $\leq p_1 + p_3 + p_4 \leq \frac{1}{q} + \frac{2}{3} = \frac{2q+3}{3q} \leq \frac{2q+3}{2q+2} = C_9(q)$ for $q \geq 2$.

Five jobs. If Balanced-LPT assigns at least one of the jobs J_3 and J_4 to the slow machine, SONL $\leq p_1 + p_3 + p_5 = P - p_2 - p_4 \leq 1 + \frac{1}{q} - 2p_5$. Now, by Corollary 1, SONL $> C_9(q)$ implies SONL $< 1 + \frac{1}{q} - 2(C_9(q) - 1)\frac{q+1}{q} = 1 + \frac{1}{q} - \frac{2}{2q+2}\frac{q+1}{q} = 1$, which contradicts the assumption that the optimal makespan is 1.

Otherwise, SONL $\leq q(p_2 + p_5)$. Since OPT runs J_2, J_3, J_4, and J_5 on the fast machine, $p_3 + p_4 + p_5 \leq 1 - p_2$. Thus, $p_5 \leq \frac{1}{3}(1 - p_2)$, and SONL $\leq q(p_2 + p_5) \leq \frac{q}{3}(1 + 2p_2)$. Furthermore, SONL $\leq P - p_2 \leq 1 + \frac{1}{q} - p_2$. Combining these two upper bounds on SONL, we obtain the inequality $(\frac{3}{q} + 2)$SONL $\leq (1 + 2p_2) + (2 + \frac{2}{q} - 2p_2) = 3 + \frac{2}{q}$, which is equivalent to SONL $\leq \frac{3q+2}{2q+3} = C_{10}(q)$.

Note that since the analysis is valid for all of interval 9, this means that Balanced-LPT has optimal competitive ratio for $2 \leq q \leq q_{\mathrm{LPT}} \approx 2.04$, as well as LPT.

Intervals 13 and 14. Recall that $C_{13}(q) = \frac{3}{4} + \frac{1}{q}$ and $C_{14}(q) = 1 + \frac{q^2+2q-2-\sqrt{q^4+8q+4}}{2q+4}$, and note that, in intervals 13 and 14, $C(q) = \max\{C_{13}(q), C_{14}(q)\}$.

By Lemma 3, for any sequence such that SONL $> C(q)$, OPT runs at most one job on M_q and at most 4 jobs on M_1, since

$$\begin{aligned}
C_{13}(q) &> 1 + \frac{1}{2q+2}, \text{ for } q < \frac{1}{2}(1+\sqrt{17}) \approx 2.56,\\
C_{13}(q) &> 1 + \frac{q}{5q+5}, \text{ for } q < \frac{1}{6}(5+\sqrt{105}) \approx 2.54,\\
C_{14}(q) &> 1 + \frac{1}{2q+2}, \text{ for } q \geq 2.491, \text{ and}\\
C_{14}(q) &> 1 + \frac{q}{5q+5}, \text{ for } q \geq 2.49.
\end{aligned}$$

Two jobs. If Balanced-LPT assigns J_2 to M_1, $qp_2 > C(q)(p_1+p_2) > p_1+p_2$ by the definition of the algorithm, so OPT $= p_1 + p_2 =$ SONL. Otherwise, SONL $= qp_2 \leq C(q)(p_1 + p_2) \leq C(q)$OPT.

Before turning to sequences with exactly three jobs, we make some general observations on sequences of more than two jobs.

Let J_i denote the first job assigned to M_q by Balanced-LPT. By the definition of the algorithm, $i > 1$. If J_2 runs on M_1, then J_3 runs on M_q, since $qp_3 < 3p_3 \leq p_1 + p_2 + p_3$. Thus, $J_i = J_2$ or $J_i = J_3$.

Let P_1 be the total size of jobs that OPT assigns to M_1. By Lemma 1, SONL $\leq P_1 + \frac{1}{q} - p_i$, if the sequence has at least $i + 1$ jobs. If SONL $> C(q) \geq C_{13}(q) = \frac{3}{4} + \frac{1}{q}$, this gives

$$p_i < P_1 - \frac{3}{4} \leq \frac{1}{4}P_1, \tag{1}$$

since $P_1 \leq 1$.

Now, let J_k be the largest job that OPT runs on M_1. Since OPT runs at most 4 jobs on M_1, $p_k \geq \frac{1}{4}P_1 > p_i$, and hence J_i arrives later than J_k, i.e., $i > k$. Thus, if Balanced-LPT runs J_2 on M_q, then OPT puts J_1 on M_1. But then the largest job on M_q in OPT's schedule is no larger than p_2, so OPT $\geq P - p_2$, and, by Lemma 1, SONL $\leq P - p_2 \leq$ OPT. Therefore, in the remaining part of the analysis, we will only consider the case where Balanced-LPT runs J_1 and J_2 on M_1 and J_3 on M_q.

Three jobs. By Lemma 1, SONL $= qp_3$. If OPT runs all jobs on M_1, OPT $= p_1 + p_2 + p_3 \geq 3p_3 > qp_3$. Otherwise OPT $\geq qp_3$.

Four jobs. Since we consider the case where J_2 is assigned to M_1, $qp_2 > C(q)(p_1+p_2)$, by the definition of Balanced-LPT. Thus, SONL $\le p_1+p_2+p_4 \le \frac{q}{C(q)}p_2 + p_4$.

If OPT runs both J_1 and J_2 on M_1, OPT $\ge p_1+p_2+p_4 \ge$ SONL.

Otherwise, OPT runs J_1 or J_2 on M_q, meaning that $p_2 \le \frac{1}{q} < \frac{1}{C(q)}$. Assuming SONL $>$ $C(q)$, inequality (1) yields $p_4 \le p_3 < \frac{1}{4}$. Substituting these upper bounds on p_2 and p_4 in the upper bound on SONL, we get SONL $< \frac{1}{C(q)} + \frac{1}{4}$. Hence, for $C(q) = C_{13}(q)$, SONL $> C(q)$ implies $q > \frac{1}{5}(5+\sqrt{65}) \approx 2.61$ and for $C(q) = C_{14}(q)$, we get $q \le 2.45$. Both are contradictions.

Five jobs. We first consider the case where Balanced-LPT assigns J_4 to the slow machine. In this case, Lemma 1 gives that SONL $\le \frac{q+1}{q} - p_3 - p_4 \le \frac{q+1}{q} - 2p_5$. Using Corollary 1, we get

$$\text{SONL} \;<\; \frac{q+1}{q} - 2(C(q)-1)\frac{q+1}{q} \;=\; \frac{q+1}{q}(3-2C(q)).$$

For $C(q) = C_{13}(q)$, this is smaller than $C(q)$ as long as $3q^2 - 6q - 8 \ge 0$, which is true when $q \le \frac{1}{3}(3+\sqrt{33}) \approx 2.92$. For $C(q) = C_{14}(q)$, the upper bound is smaller than $C(q)$ whenever q is at least the largest root of $9q^4 - 9q^3 - 28q^2 - 5q + 6$. This root is smaller than 2.37.

Thus, the case left to consider is that Balanced-LPT puts J_1, J_2, and J_4 on the fast machine and J_3 on the slow machine. We assume that SONL $> C(q)$ and show that this leads to a contradiction.

First, SONL $> C(q)$ implies the following three inequalities, where inequality (4) holds by Corollary 1.

$$q(p_3+p_5) \;>\; C(q) \tag{2}$$

$$p_1+p_2+p_4+p_5 \;>\; C(q) \tag{3}$$

$$p_5 \;>\; (C(q)-1)\frac{q+1}{q} \tag{4}$$

Furthermore, since OPT runs four jobs on M_1,

$$p_2+p_3+p_4+p_5 \;\le\; 1 \tag{5}$$

Since Balanced-LPT runs J_2 on M_1, by the definition of the algorithm,

$$p_1+p_2 \;<\; \frac{q}{C(q)}\,p_2 \tag{6}$$

- Finally, J_4 is at least as large as J_5:

$$p_4 \ge p_5 \tag{7}$$

We now find a linear combination of inequalities (2)–(7) eliminating all variables p_1, ..., p_5; adding up $\frac{1}{C(q)}$ times (2), (3), $(\frac{q}{C(q)} - 2)$ times (4) (note that $C(q) < \frac{q}{2}$ in the current intervals), $-\frac{q}{C(q)}$ times (5), -1 times (6), and $(\frac{q}{C(q)} - 1)$ times (7), we arrive at the inequality

$$0 > -(q+2)C^2(q) + (q^2 + 4q + 2)C(q) - (2q^2 + q).$$

This is true for $C(q) < 1 + \frac{q^2+2q-2-\sqrt{q^4+8q+4}}{2q+4} = C_{14}(q)$ and for $C(q) > 1 + \frac{q^2+2q-2+\sqrt{q^4+8q+4}}{2q+4}$ which is greater than 2 for $q \geq 2$. Both possibilities are contradictions, the latter because $C(q) \leq \frac{1}{4}(1 + \sqrt{17}) \approx 1.28$.

4.3 Intervals 11 and 12

When assigning the second job, the following algorithm tries to do the opposite of LPT. If $qp_2 < p_1 + p_2$, LPT puts J_2 on M_q, so Opposite-LPT puts J_2 on M_1, unless $p_1 + p_2 > C(q)qp_2$. Similarly, if $qp_2 \geq p_1 + p_2$, Opposite-LPT puts J_2 on M_q. Note that it is not necessary to check that $qp_2 \leq C(q)(p_1 + p_2)$, as argued below.

Algorithm Opposite-LPT

Assign J_1 to M_1.

If $qp_2 < p_1 + p_2 \leq C(q)\, qp_2$, assign J_2 to M_1, and otherwise to M_q.

Assign the remaining jobs by the LPT rule.

Recall that $C_{11}(q) = \frac{1}{6q}(q^2 + 3 + \sqrt{q^4 - 6q^2 + 24q + 9})$ and $C_{12}(q) = \frac{q}{2}$, and note that in intervals 11 and 12, $C(q) = \max\{C_{11}(q), C_{12}(q)\}$.

By Lemma 3, if SONL $> C(q)$, OPT runs at most four jobs on M_1, since in intervals 11 and 12, $C(q) \geq C_{11}(q) > 1 + \frac{q}{5q+5}$ for $q > 0$, and at most one job on M_q, since $C_{11}(q) > 1 + \frac{1}{2q+2}$ for $q \geq 2.02$. Thus, we need only consider sequences with up to five jobs, where OPT places exactly one job on the slow machine.

Balanced-LPT

Two jobs. It is always safe to put the second job on M_q, since $C(q) \leq \frac{1}{2}q$, i.e., $qp_2 \leq \frac{1}{2}q(p_1 + p_2) \leq C(q)(p_1 + p_2)$.

Furthermore, if Opposite-LPT puts J_2 on M_1 then, by the definition of the algorithm, $p_1 + p_2 \leq C(q)\, qp_2 \leq C(q)\,\mathrm{OPT}$.

For sequences of more than two jobs, the proof is split in two cases according to where Opposite-LPT schedules J_2.

Case A: Opposite-LPT schedules J_2 on M_q.

The analysis of this case is similar to the analysis of Intervals 9 and 10.
As in Intervals 9 and 10 we need only consider the case where the job that OPT runs on M_q is J_1, because otherwise OPT $\geq P - p_2 \geq$ SONL.

Three jobs. The proof is the same as for Intervals 9 and 10: Since we consider the case where OPT runs J_1 on M_q and J_2 and J_3 on M_1, OPT $= \max\{qp_1, p_2 + p_3\}$. Since Opposite-LPT runs J_2 on M_q, by Lemma 1, SONL $\leq p_1 + p_3 \leq 2p_1 \leq qp_1 \leq$ OPT.

Four jobs. We use the same reasoning as in intervals 9 and 10: Since $p_2 + p_3 + p_4 \leq 1$ and $p_2 \geq p_3 \geq p_4$, we get $p_3 + p_4 \leq \frac{2}{3}$. Thus, using Lemma 1 and $p_1 \leq \frac{1}{q}$, we have SONL $\leq p_1 + p_3 + p_4 \leq \frac{1}{q} + \frac{2}{3}$, which is smaller than $C(q)$ in intervals 11 and 12.

Five jobs. We split the proof in subcases according to what caused Opposite-LPT to put J_2 on M_q.
Case 1: $p_1 + p_2 > C(q) \cdot qp_2$.

In this case, $p_2 < \frac{1}{qC(q)-1} p_1 \leq \frac{1}{qC(q)-1} \frac{1}{q}$.
Since OPT schedules J_1 on M_q and the remaining jobs on M_1, $p_1 \leq \frac{1}{q}$ and $p_3 + p_5 \leq \frac{1}{2}(p_2 + p_3 + p_4 + p_5) \leq \frac{1}{2}$. Thus, if Opposite-LPT schedules J_3 or J_4 on M_q, SONL $\leq p_1 + p_3 + p_5 \leq \frac{1}{q} + \frac{1}{2} < 1$. But this would contradict the assumption that the best possible schedule has a makespan of 1. Hence, Opposite-LPT must schedule both J_3 and J_4 on M_1.
Thus,

$$\text{SONL} \leq q(p_2 + p_5) \leq q\left(p_2 + \frac{1 - p_2}{3}\right) = \frac{2q}{3}p_2 + \frac{q}{3} < \frac{2}{3qC(q) - 3} + \frac{q}{3},$$

which is smaller than $C(q)$ in interval 12 and equal to $C(q)$ in interval 11.

Case 2: $p_1 + p_2 \leq qp_2$.

In this case $p_2 \geq \frac{p_1}{q-1}$. Thus, we get

$$\begin{aligned}
\text{SONL} &\leq P - p_2 \\
&\leq 1 + p_1 - p_2, && \text{by } p_2 + p_3 + p_4 + p_5 \leq 1 \\
&\leq 1 + p_1\left(1 - \frac{1}{q-1}\right), && \text{by } p_2 \geq \frac{p_1}{q-1} \\
&\leq 1 + \frac{q-2}{q^2 - q}, && \text{by } p_1 \leq \frac{1}{q} \text{ and } q \geq 2 \\
&= \frac{q^2 - 2}{q^2 - q},
\end{aligned}$$

which is smaller than both $C_{11}(q)$ and $C_{12}(q)$ for $q > 1$.

Case B: Opposite-LPT schedules J_2 on M_1.

By the definition of the algorithm, $qp_2 < p_1 + p_2 \le C(q)\, qp_2$. Furthermore, Opposite-LPT places J_3 on M_q, since $p_1 + p_2 + p_3 \ge 3p_3 > qp_3$.

Three jobs. Since, J_3 determines the makespan, SONL $= qp_3$. However, OPT $\ge \min\{qp_3, p_1 + p_2 + p_3\} = qp_3 =$ SONL.

Four jobs. Since Opposite-LPT puts J_2 on M_1, $qp_2 \le p_1 + p_2$, which is equivalent to $p_2 \le \frac{p_1}{q-1}$.

We analyze three cases according to which job is run on M_q by OPT.

Case 1: OPT runs J_3 or J_4 on M_q.

In this case, OPT $\ge p_1 + p_2 + p_4$. By Lemma 1, SONL $\le p_1 + p_2 + p_4 \le$ OPT.

Case 2: OPT runs J_1 on M_q.

This gives $p_2 + p_3 + p_4 \le 1$ and $p_1 \le \frac{1}{q}$. Thus,

$$\begin{aligned} \text{SONL} &\le p_1 + p_2 + p_4 \le p_1 + p_2 + \frac{1 - p_2}{2} = p_1 + \frac{1}{2}p_2 + \frac{1}{2} \\ &\le p_1 + \frac{1}{2} \cdot \frac{p_1}{q-1} + \frac{1}{2} = \frac{2q-1}{2q-2}p_1 + \frac{1}{2} \le \frac{2q-1}{2q(q-1)} + \frac{1}{2}, \end{aligned}$$

which is smaller than $C_{11}(q)$ in interval 11 and smaller than $C_{12}(q)$ in interval 12.

Case 3: OPT runs J_2 on M_q.

In this case, $p_2 \le \frac{1}{q}$ and $p_1 + p_3 + p_4 \le 1$.

We have two upper bounds on the makespan of Opposite-LPT; SONL $\le p_1 + p_2 + p_4$ and SONL $\le q(p_3 + p_4) \le q(1 - p_1)$. We use the first upper bound on SONL to derive a lower bound on p_1:

$$\begin{aligned} \text{SONL} &\le p_1 + p_2 + p_4 \le p_1 + p_2 + \frac{1 - p_1}{2} = \frac{1}{2}p_1 + p_2 + \frac{1}{2} \\ &\le \frac{1}{2}p_1 + \frac{p_1}{q-1} + \frac{1}{2} = \frac{q+1}{2(q-1)}p_1 + \frac{1}{2} \end{aligned}$$

Assume for the sake of contradiction that SONL $> C(q)$. Then, by the inequality above, $\frac{q+1}{2(q-1)}p_1 + \frac{1}{2} > C(q)$, which is equivalent to

$$p_1 > \frac{(2C(q) - 1)(q-1)}{q+1}.$$

Thus,

$$\text{SONL} \le q(1 - p_1) \le q\left(1 - \frac{(2C(q) - 1)\,(q-1)}{q+1}\right),$$

which can be simplified to $C(q) < \frac{2q^2}{2q^2-q+1}$. For $C(q) = C_{11}(q)$, this holds only for $q < 0$, and for $C(q) = C_{12}(q)$, it holds only for $q < 2.28 < q_{11}$. Thus, we arrived at a contradiction.

Five jobs. If Opposite-LPT runs J_4 on M_1, then

$$\begin{aligned}\text{SONL} &\le q(p_3+p_5)\\ &\le \frac{q}{2}(p_2+p_3+p_4+p_5),\ \text{since } p_3 \le p_2 \text{ and } p_5 \le p_4\\ &\le \frac{q}{2},\ \text{since OPT schedules four jobs on } M_1\\ &\le C(q)\,.\end{aligned}$$

If Opposite-LPT runs J_4 on M_q, then $q(p_3+p_4) \le p_1+p_2+p_4$. Assume for the sake of contradiction that SONL $> C(q)$. Then, by Corollary 1, $p_5 > (C(q)-1)\frac{q+1}{q}$, so

$$p_1+p_2+p_4 \ \le\ 1+\frac{1}{q}-(p_3+p_5) \ <\ \frac{q+1}{q}-2(C(q)-1)\frac{q+1}{q} \ =\ (3-2C(q))\frac{q+1}{q}$$

and

$$q(p_3+p_4) \ \ge\ 2qp_5 \ >\ 2(C(q)-1)(q+1)\,.$$

Since $2(C(q)-1) \ge \frac{(3-C(q))}{q}$ in Intervals 11 and 12, this contradicts $q(p_3+p_4) \le p_1+p_2+p_4$.

5 Matching Lower Bounds

In this section we present job sequences that prove the lower bounds matching the upper bounds of Section 4 or, in the intervals where LPT is optimal, the bounds of LPT as given in [7]. In all sequences, unless otherwise mentioned, jobs are scaled so that if the sequence is completed, OPT $= 1$. All sequences have between three and six jobs, most of them have exactly five jobs.

Interval 1 ($1 \le q \le q_1 \approx 1.04$): $C_1(q) = \frac{2}{3}+\frac{1}{2q}$.
The sequence consists of five jobs with sizes

$$p_1 = p_2 = \frac{1}{2q},\ \ p_3 = p_4 = p_5 = \frac{1}{3}.$$

The schedule of OPT_2 is achieved by running one job on each machine. This gives $\text{OPT}_2 = \frac{1}{2}$. If SONL schedules both jobs on M_1, $\text{SONL}_2 = \frac{1}{q} = \frac{2}{q}\cdot\text{OPT}_2 > C_1(q)\cdot\text{OPT}_2$, and assigning both to M_q only makes the competitive ratio worse. Thus, we need only consider algorithms that put exactly one of the first two jobs on M_q.

For the complete sequence, OPT runs the first two jobs on M_q and the other jobs on M_1. If SONL assigns two of the last three jobs to M_1, SONL $\geq \frac{1}{2q}+\frac{2}{3}$, and if SONL assigns two of the last three jobs to M_q, SONL $\geq \frac{1}{2}+\frac{2}{3}q \geq \frac{1}{2q}+\frac{2}{3}$.

In the next two intervals, any algorithm putting the first job on the fast machine has a competitive ratio of at least $C(q)$. The sequence proving this is based on a sequence from [7]. The definition of the sequence (as a function of p_1) is

$$p_2 = \frac{3+2q-2q^2}{2q^2+q}p_1,\ p_3 = p_4 = p_5 = \frac{q+1}{2q+1}p_1.$$

For $1 \leq q \leq \frac{1}{2}(1+\sqrt{7})$, $0 \leq \frac{3+2q-2q^2}{2q^2+q} \leq 1$, so the sequence is well-defined and $p_2 \leq p_1$. Furthermore, since $q \leq \frac{1}{6}(1+\sqrt{37}) \approx 1.18$, $p_3 \leq p_2$.

Putting J_1 on M_1 and J_2 on M_q, J_2 finishes before J_1, since $\frac{3+2q-2q^2}{2q^2+q}\,q \leq 1$ for $q \geq 1$. Thus, $\mathrm{OPT}_2 = p_1$. Furthermore, scheduling the first two jobs on M_q and the last three on M_1 gives a makespan of $q(p_1+p_2) = p_3+p_4+p_5 = \frac{3q+3}{2q+1}p_1$. Thus, $\mathrm{OPT} = \frac{3q+3}{2q+1}p_1$.

Assume that SONL puts J_1 on M_1. Then, if it also schedules J_2 on M_1,

$$\frac{\mathrm{SONL}_2}{\mathrm{OPT}_2} = \frac{p_1+p_2}{p_1} = \frac{3q+3}{2q^2+q} > \frac{3}{2q} \geq \frac{3}{2}\sqrt{\frac{3}{4}} > 1.299 > C(q)\,,$$

and if it schedules J_2 on M_q,

$$\mathrm{SONL} \geq q(p_2+2p_3) = p_1+2p_3 = \frac{4q+3}{2q+1}\,p_1 = \frac{4q+3}{3q+3}\,\mathrm{OPT}\,.$$

In Intervals 2 and 3, $\frac{4q+3}{3q+3} > C(q)$.

This shows that any algorithm scheduling J_1 on M_1 has a competitive ratio of at least $C(q)$ in Intervals 2 and 3. Thus, only the case where SONL schedules J_1 on M_q is left to consider in each of those two intervals.

Interval 2 ($q_1 \leq q \leq q_2 \approx 1.14$): $C_2(q) = 1+\frac{1}{2}\big(4q^2+4q-1-\sqrt{(4q^2+4q-1)^2-4q^2}\big)$. The sequence consists of five jobs with sizes

$$p_1 = \frac{1}{q} - \frac{2q+1}{q+1}p_5,\ \ p_2 = \frac{2q+1}{q+1}p_5,\ \ p_3 = 1-2p_5,$$
$$p_4 = p_5 = \frac{q+1}{2q}(4q^2+4q-1-\sqrt{(4q^2+4q-1)^2-4q^2}).$$

The first job is larger than the second job, since $\frac{1}{q} > \frac{4q+2}{q+1}p_5$ for $q < \frac{1}{4}(1+\sqrt{13}) \approx 1.15$. This is equivalent to $p_5 < \frac{q+1}{4q^2+2q}$ for $q < \frac{1}{4}(1+\sqrt{13})$. Since $\frac{q+1}{4q^2+2q} \le \frac{1}{3}$ for $q \ge 1$, this also implies $p_3 \ge p_4$. Finally, $p_2 \ge p_3$, since $p_5 \ge \frac{q+1}{4q+3}$ for $q \le \frac{1}{4}(1+\sqrt{13})$.

Case 1: SONL assigns J_2 to M_q.

In this case, $\mathrm{SONL}_2 = q(p_1+p_2) = 1$.

Clearly, $\mathrm{OPT}_2 \le \max\{p_1, qp_2\}$. By the proof that $p_1 \ge p_2$, $p_2 \le \frac{1}{2q}$. Thus, $qp_2 \le \frac{1}{2}$. Furthermore, $p_3 \le p_2 < \frac{1}{2}$, so $p_5 \ge \frac{1}{2}(1-p_3) > \frac{1}{4}$. Hence,

$$p_1 \;=\; \frac{1}{q} - \frac{2q+1}{q+1}p_5 \;<\; \frac{1}{q} - \frac{2q+1}{4q+4} \;\le\; \frac{2}{3}\,, \text{ for all } q \ge 1.$$

This shows that $\mathrm{SONL} \ge \frac{3}{2}\,\mathrm{OPT} > C(q)\,\mathrm{OPT}$.

algorithms.

Case 2: SONL assigns J_2 to M_1 and J_3 to M_q.

In this case,

$$\mathrm{SONL}_3 \ge (p_1+p_3)q = 1 - \frac{2q^2+q}{q+1}\,p_5 + q - \frac{2q^2+2q}{q+1}\,p_5 = q+1 - \frac{4q^2+3q}{q+1}\,p_5\,.$$

On the other hand,

$$\mathrm{OPT}_3 \le \max\{qp_1, p_2+p_3\} = \max\left\{1 - \frac{2q^2+q}{q+1}\,p_5,\; 1 - \frac{1}{q+1}\,p_5\right\} = 1 - \frac{1}{q+1}\,p_5\,.$$

Thus, in this case $\mathrm{SONL}_3 < C(q)\mathrm{OPT}_3$, if and only if $q+1-\frac{4q^2+3q}{q+1}p_5 < C(q)(1-\frac{1}{q+1}p_5)$. Note that $p_5 = \frac{1}{q}(C(q)-1)(q+1)$. Substituting this in the inequality, we get $C(q) < 1+\frac{1}{2}\big(4q^2+4q-1+\sqrt{(4q^2+4q-1)^2-4q^2}\big) = C_2(q)$, which is a contradiction.

Case 3: SONL assigns J_2 and J_3 to M_1.

Consider the whole sequence. OPT runs J_1 and J_2 on M_q and the other jobs on M_1. If SONL runs J_4 and J_5 on M_q,

$$\mathrm{SONL} \;\ge\; (p_1+2p_5)q \;=\; \left(\frac{1}{q} - \frac{2q+1}{q+1}p_5 + \frac{2q+2}{q+1}p_5\right)q \;=\; 1 + \frac{q}{q+1}p_5 \;=\; C_2(q).$$

Otherwise,

$$\mathrm{SONL} \;\ge\; p_2+p_3+p_5 \;=\; \frac{2q+1}{q+1}p_5 + 1 - \frac{q+1}{q+1}p_5 \;=\; 1 + \frac{q}{q+1}p_5 \;=\; C_2(q).$$

Interval 3 ($q_2 \le q \le \sqrt{4/3} \approx 1.15$): $C_3(q) = \frac{6q+4}{3q+6}$.
This is the only case of a lower bound sequence that consists of six jobs. Let $\beta = \frac{4-3q^2}{3q(q+2)}$. Note that, in this interval, $\beta > 0$. The sequence is

$$p_1 = 1 - 2p_5, \quad p_2 = p_3 = p_4 = \frac{1}{3q}, \quad p_5 = p_6 = \frac{1}{3q} - \beta.$$

The first job is larger than the second job, because $p_1 = 1 - 2p_5 > 1 - 2p_2 > p_2$, since $p_2 < \frac{1}{3}$.

First note that

$$\begin{aligned} \mathrm{OPT}_4 \;&\le\; \max\{p_1 + p_2,\, q(p_3 + p_4)\} \;=\; \max\left\{1 - 2\left(\frac{1}{3q} - \beta\right) + \frac{1}{3q},\, \frac{2}{3}\right\} \\ &=\; \max\left\{1 - \frac{1}{3q} + 2\beta,\, \frac{2}{3}\right\} \;=\; 1 - \frac{1}{3q} + 2\beta \;=\; \frac{3q(q+2) - (q+2) + 2(4-3q^2)}{3q(q+2)} \\ &=\; \frac{6 + 5q - 3q^2}{3q(q+2)}. \end{aligned}$$

If SONL puts all three jobs J_2, J_3, and J_4 on M_1,

$$\mathrm{SONL}_4 \;\ge\; \frac{1}{q} \;=\; \frac{6 + 5q - 3q^2}{3q+6}\,\mathrm{OPT}_4 \;>\; C_3(q)\,\mathrm{OPT}_4, \;\text{ for } q < \frac{1}{6}(5 + \sqrt{97}) \approx 2.47.$$

If SONL puts exactly one of the jobs J_2, J_3, and J_4 on M_1,

$$\mathrm{SONL}_4 \ge q(p_1 + p_3 + p_4) = q(1 - 2(\frac{1}{3q} - \beta) + \frac{2}{3q}) = q(1 + 2\beta) > 1,$$

yielding an even worse ratio.

We finally consider the case that SONL puts exactly two of the jobs J_2, J_3, and J_4 on M_1. If J_5 and J_6 are both scheduled on M_1,

$$\mathrm{SONL} \;\ge\; p_3 + p_4 + p_5 + p_6 \;=\; \frac{2}{3q} + \frac{2}{3q} - 2\beta \;=\; \frac{4(q+2) - 2(4-3q^2)}{3q(q+2)} \;=\; \frac{6q+4}{3q+6}.$$

Otherwise,

$$\mathrm{SONL} \;\ge\; q(p_1 + p_4 + p_5) \;=\; q(1 - 2p_5 + p_4 + p_5) \;=\; q(1 + p_4 - p_5) \;=\; q(1 + \beta) \;=\; \frac{6q+4}{3q+6}.$$

In the remaining intervals, $q \ge C(q)$, so any algorithm scheduling the first job on M_q has a competitive ratio of at least $C(q)$. Thus, only the case where SONL schedules J_1 on M_1 needs to be analyzed.

For intervals 4–9, we use sequences given in [7] as negative examples for LPT. We show that those sequences are in fact lower bound sequences for any semi-online algorithm. In intervals 4 and 9, the proof holds for the entire interval, even though LPT is not optimal in the complete interval.

Intervals 4 and 5 ($\sqrt{4/3} \le q \le \frac{1}{4}(1+\sqrt{17}) \approx 1.28$ *and* $\frac{1}{4}(1+\sqrt{17}) \le q \le \sqrt{2} \approx 1.41$*)* : $C_4(q) = q$ and $C_5(q) = \frac{1}{2} + \frac{1}{q}$.

The sequence consists of three jobs with sizes

$$p_1 = \frac{1}{q}, \;\; p_2 = p_3 = \frac{1}{2}.$$

The optimal makespan of 1 is achieved by running J_1 on M_q and J_2 and J_3 on M_1.

If SONL assigns both p_2 and p_3 to M_q, SONL $\ge q$. Otherwise, SONL $\ge \frac{1}{q} + \frac{1}{2}$.

In Interval 4, $\frac{1}{q} + \frac{1}{2} \ge q$, so SONL $\ge q$. In Interval 5, $\frac{1}{q} + \frac{1}{2} \le q$, so SONL $\ge \frac{1}{q} + \frac{1}{2}$.

Intervals 6 and 9 ($\sqrt{2} \le q \le \frac{1}{4}(1+\sqrt{33}) \approx 1.69$ and $2 \le q \le \frac{1}{2}(1+\sqrt{11}) \approx 2.16$) $C_6(q) = C_9(q) = 1 + \frac{1}{2q+2}$.
The sequence consists of four jobs with sizes

$$p_1 = \frac{2q^2+q-2}{2q(q+1)}, \;\; p_2 = \frac{q+2}{2q(q+1)}, \;\; p_3 = p_4 = \frac{1}{2q}.$$

Since $q \ge \sqrt{2}$, $p_2 \le p_1$. Moreover, it is easy to see that $p_3 \le p_2$ for all $q \ge 1$.

OPT_2 is achieved by running J_1 on M_1 and J_2 on the slow machine. In Interval 6, $qp_2 \ge p_1$, so $\mathrm{OPT}_2 = qp_2$. In Interval 9, $qp_2 \le p_1$, so $\mathrm{OPT}_2 = p_1$.

If the algorithm assigns J_2 to M_1, we get $\mathrm{SONL}_2 = p_1 + p_2$. In Interval 6, this gives a ratio of

$$\frac{\mathrm{SONL}_2}{\mathrm{OPT}_2} = \frac{p_1+p_2}{qp_2} = \frac{2q+2}{q+2} = 1 + \frac{q}{q+2} \ge 1 + \frac{1}{2q+2} = C_6(q), \;\text{ for } q \ge 1.$$

In Interval 9, it gives a ratio of

$$\frac{\mathrm{SONL}_2}{\mathrm{OPT}_2} = \frac{p_1+p_2}{p_1} = 1 + \frac{q+2}{2q^2+q-2} \ge 1 + \frac{1}{2q+2} = C_9(q), \;\text{ for } q > 0.$$

We now turn to the case where SONL assigns J_2 to M_q. If SONL runs at least one of the jobs J_3 and J_4 on M_q, then

$$\mathrm{SONL} \ge q(p_2+p_4) = \frac{2q+3}{2q+2} = C_6(q) = C_9(q).$$

Otherwise, all jobs but J_2 run on M_1, and

$$\mathrm{SONL} \ge p_1 + p_3 + p_4 = \frac{2q^2+3q}{2q(q+1)} = \frac{2q+3}{2q+2} = C_6(q) = C_9(q).$$

The optimal makespan is achieved by running the first two jobs on M_1 and the last two jobs on M_q.

Interval 7 ($\frac{1}{4}(1+\sqrt{33}) \le q \le \frac{1}{2}(1+\sqrt{7}) \approx 1.82$): $C_7(q) = \frac{2q+1}{q+2}$. The sequence consists of four jobs with sizes

$$p_1 = \frac{1}{q} = \frac{q+2}{q(q+2)}, \;\; p_2 = \frac{2+2q-q^2}{q(q+2)}, \;\; p_3 = p_4 = \frac{q^2-1}{q(q+2)}.$$

Since $q \ge 1$, $p_1 \ge p_2$. Moreover, $p_2 \ge p_3$, since $2q^2 - 2q - 3 \le 0$ holds for $q \le \frac{1}{2}(1+\sqrt{7})$.

OPT_2 is achieved by running J_1 on M_1 and J_2 on M_q. In this interval, $qp_2 \ge p_1$, since $q+2 \le q(2+2q-q^2)$ for $1 \le q \le 2$, so $\mathrm{OPT}_2 = qp_2$.

Any algorithm assigning J_2 to M_1 has a competitive ratio of at least $C_7(q)$, since $\frac{p_1+p_2}{qp_2} \ge \frac{2q+1}{q+2}$ when $2q^4 - 4q^3 - 5q^2 + 8q + 8 \ge 0$, and this latter inequality is true throughout the interval. Thus, we turn to algorithms assigning J_2 to M_q.

For the complete sequence, OPT runs J_1 on M_q, and all other jobs on M_1. If the semi-online algorithm runs at least one more job than J_2 on M_q, it gives a load of at least $q(p_2+p_3) = \frac{2q+1}{q+2}$. Otherwise, the load on M_1 is at least $p_1 + p_3 + p_4 = \frac{2q^2+q}{q(q+2)} = \frac{2q+1}{q+2}$. In both cases, the lower bound on the competitive ratio is achieved.

Interval 8 ($\frac{1}{2}(1+\sqrt{7}) \le q \le 2$): $C_8(q) = \frac{2}{3} + \frac{1}{q}$.
The sequence consists of four jobs with sizes

$$p_1 = \frac{1}{q}, \;\; p_2 = p_3 = p_4 = \frac{1}{3}.$$

OPT runs J_1 on M_q, and the other jobs on M_1. If SONL runs at least two jobs on M_q, $\mathrm{SONL} \ge \frac{2q}{3} \ge \frac{2}{3} + \frac{1}{q}$, for $q \ge \frac{1}{2}(1+\sqrt{7})$. Otherwise, $\mathrm{SONL} \ge \frac{1}{q} + \frac{2}{3}$. In both cases the lower bound on the competitive ratio is achieved.

Interval 10 ($\frac{1}{2}(1+\sqrt{11}) < q \le q_{10} \approx 2.20$): $C_{10}(q) = \frac{3q+2}{2q+3}$.
The sequence consists of five jobs:

$$p_1 = \frac{1}{q}, \;\; p_2 = \frac{-q^2+3q+3}{2q^2+3q}, \;\; p_3 = p_4 = p_5 = \frac{q^2-1}{2q^2+3q} = \frac{1}{3}(1-p_2).$$

The sequence is non-increasing, since $p_1 \ge p_2$ for all $q \ge 1$ and $p_2 \ge p_3$ for $q \le \frac{1}{4}(3+\sqrt{41})$.

Consider the subsequence J_1, J_2. Since $q^3 - 3q^2 - q + 3 \le 0$ for $1 \le q \le 3$, $\mathrm{OPT}_2 = \max\{p_1, qp_2\} = qp_2$. If SONL puts J_2 on M_1, $\mathrm{SONL}_2 = \frac{-q^2+5q+6}{2q^2+3q}$. This is larger than $C_{10}(q) \cdot qp_2$ for $q \le q_{10}$.

Consider now the case, where SONL puts J_2 on M_q. If J_2 is the only job to be put on M_q,

$$\mathrm{SONL} \;\ge\; p_1 + 3p_3 \;=\; \frac{2q+3+3q^2-3}{2q^2+3q}\frac{3q+2}{2q+3} \;=\; C_{10}(q).$$

If J_2 is not alone on M_q,

$$\text{SONL} \;\geq\; q(p_2+p_3) \;=\; q\frac{3q-q^2+3+q^2-1}{2q^2+3q} \;=\; C_{10}(q).$$

Since $qp_1 = p_2 + 3p_3 = 1$, this proves the lower bound.

Interval 11 ($q_{10} \leq q \leq q_{11} \approx 2.33$): $C_{11}(q) = \frac{1}{6q}(q^2+3+\sqrt{q^4-6q^2+24q+9})$.
The sequence consists of five jobs:

$$p_1 = \frac{1}{q},\;\; p_2 = \frac{1}{4q^2}(3-q^2+\sqrt{q^4-6q^2+24q+9}),\;\; p_3 = p_4 = p_5 = \frac{1}{3}(1-p_2).$$

To show that the sequence is valid we need to show $p_1 \geq p_2$ and $p_2 \geq p_3$. The first is true for all $q \geq \sqrt{7}-1 \approx 1.65$. The second results in the inequality $q^3-2q-8 \leq 0$ which is true for all $q \leq q_{11}$.

Consider the subsequence J_1, J_2. Since $qp_2 \geq \frac{1}{q}$ is equivalent to $q^2-3q-1 \leq 0$, which holds for $q \leq \frac{1}{2}(3+\sqrt{13})$, $\text{OPT}_2 = qp_2$. If SONL puts J_2 on M_1, $\text{SONL}_2 = p_1+p_2$ which gives the ratio $\frac{p_1}{qp_2}+\frac{1}{q}$ that is exactly equal to $C_{11}(q)$. Thus, we turn to the case where SONL puts J_2 on M_q. If at least one other job is put on M_q, $\text{SONL} \geq q(p_2+\frac{1-p_2}{3}) = \frac{q}{3}(2p_2+1) = C_{11}(q)$. Otherwise, $\text{SONL} = p_1+1-p_2 = \frac{1}{q}+1-p_2 \geq C_{11}(q)$, when q is at least q_{10} and at most the largest root of $3q^4-9q^3-8q^2+21q+18$ which is approximately 2.76.

Intervals 12 and 13 ($q_{11} \leq q \leq \frac{1}{4}(3+\sqrt{41}) \approx 2.35$ and $\frac{1}{4}(3+\sqrt{41}) \leq q \leq q_{13} \approx 2.51$): $C_{12}(q) = \frac{q}{2}$, $C_{13}(q) = \frac{3}{4}+\frac{1}{q}$.
The sequence for both intervals consists of five jobs:

$$p_1 = \frac{1}{q},\;\; p_2 = p_3 = p_4 = p_5 = \frac{1}{4}.$$

OPT runs J_1 on M_q, and all other jobs on M_1.

The algorithm needs to either run at least three of the other jobs on M_1, or at least two of them on M_q. In the first case, $\text{SONL} \geq \frac{1}{q}+\frac{3}{4}$. In the second case $\text{SONL} \geq \frac{q}{2}$. Thus, the competitive ratio is at least $\min\{\frac{1}{q}+\frac{3}{4}, \frac{q}{2}\}$. In interval 12, $\frac{q}{2} \leq \frac{3}{4}+\frac{1}{q}$, and the competitive ratio is at least $C_{12}(q)$. In interval 13, $\frac{3}{4}+\frac{1}{q} \leq \frac{q}{2}$, and the competitive ratio is at least $C_{13}(q)$.

Interval 14 ($q_{13} \leq q \leq q_{14} \approx 2.57$): $C_{14}(q) = 1+\frac{q^2+2q-2-\sqrt{q^4+8q+4}}{2q+4}$.
The sequence consists of five jobs:

$$p_1 = \frac{1}{q},\;\; p_2 = 1-\frac{1}{q}-\frac{q+2}{q+1}p_5,\;\; p_3 = \frac{1}{q}-\frac{q}{q+1}p_5,$$
$$p_4 = p_5 = \frac{q+1}{2q(q+2)}\big(q^2+2q-2-\sqrt{q^4+8q+4}\big).$$

The sequence is valid, since $p_1 \geq p_2 \geq p_3$ for $q \geq 2$, $p_3 \geq p_4$ for $q \leq q_{14}$, and $p_5 \geq 0$ for $q \geq 2$.

Case 1: SONL assigns J_2 to M_q.

In this case,

$$\frac{\mathrm{SONL}_2}{\mathrm{OPT}_2} \geq \frac{qp_2}{p_1 + p_2} = C_{14}(q).$$

Case 2: SONL assigns J_2 and J_3 to M_1.

Note that $\mathrm{OPT}_3 \leq \max\{p_1 + p_2, qp_3\} = p_1 + p_2$, since $p_1 + p_2 > qp_3$ for any $q > 2$. Thus,

$$\frac{\mathrm{SONL}_3}{\mathrm{OPT}_3} = \frac{p_1 + p_2 + p_3}{p_1 + p_2} \geq C_{14}(q), \text{ for any } q \leq 2.8\,.$$

Case 3: SONL assigns J_2 to M_1 and J_3 to M_q.

For the whole sequence, OPT runs J_1 on M_q and the remaining four jobs on M_1.

If SONL runs both J_4 and J_5 run on M_1, we get $\mathrm{SONL} \geq p_1 + p_2 + 2p_5 = 1 + \frac{q}{q+1}p_5 = C_{14}(q)$. Otherwise, $\mathrm{SONL} \geq q(p_3 + p_5) = q(\frac{1}{q} + \frac{1}{q+1}p_5) = C_{14}(q)$.

Interval 15 $(q \geq q_{14})$: $C_{15}(q) = 1 + \frac{1}{2q+1}$.

For this whole interval, LPT is optimal. However, using the negative example in [7] does not directly yield the desired general bound. We use an adaptation of that sequence.

For $q \geq 1 + \sqrt{3}$, the adapted sequence consists of the five jobs

$$p_1 = \frac{2q^2 - 2q - 3}{2q^2 + q}, \quad p_2 = \frac{1}{q}, \quad p_3 = p_4 = p_5 = \frac{q+1}{2q^2 + q}.$$

Note that $p_1 \geq p_2$ for $q \geq 1 + \sqrt{3}$ and $p_2 \geq p_3$ for $q \geq \frac{1}{4}(3 + \sqrt{41})$.

For $q < 1 + \sqrt{3}$, the above sequence does not apply, since $p_1 < p_2$. Thus, we switch the order of the first two jobs and use the sequence

$$p'_1 = p_2, \quad p'_2 = p_1, \quad p'_3 = p'_4 = p'_5 = p_3.$$

For $1 - \sqrt{3} \leq q \leq 1 + \sqrt{3}$, $p'_1 \geq p'_2$. Furthermore, $p'_2 \geq p'_3$, since $2q^2 - 3q - 4 \geq 0$ for $q \geq \frac{1}{4}(3 + \sqrt{41}) \approx 2.35$ $(< q_{14})$.

Case 1: SONL runs the second job on M_q.

After two jobs, $\mathrm{OPT}_2 \leq p_1 + p_2 = \frac{2q^2-2}{2q^2+q}$.

For $q \geq 1 + \sqrt{3}$, $\mathrm{SONL}_2 \geq qp_2 = 1$. In this case, the competitive ratio is at least $\frac{2q^2+q}{2q^2-2} \geq \frac{2q+2}{2q+1}$, since $(2q+1)^2 \geq 2q(2q+2) \geq (2q - \frac{2}{q})(2q+2)$.

For $q < 1+\sqrt{3}$, $\mathrm{SONL}_2 \geq qp'_2 = qp_1 = \frac{2q^2-2q-3}{2q+1}$. This violates the competitive ratio when $\frac{2q^2-2q-3}{2q^2-2} \geq \frac{2q+2}{2q^2+q}$. This is true when q is at least the largest root of $4q^4 - 6q^3 - 12q^2 + q + 4$.

Case 2: SONL runs the second job on M_1.

OPT is achieved by running J_2 on M_q and all other jobs on M_1. If SONL schedules at least two jobs on M_q, then SONL $\geq \frac{2q(q+1)}{q(2q+1)} = \frac{2q+2}{2q+1}$. Otherwise, two extra jobs (apart from J_1 and J_2) run on M_1, which gives SONL $\geq \frac{2q^2-2}{2q^2+q} + \frac{2(q+1)}{2q^2+q} = \frac{2q^2+2q}{2q^2+q} = \frac{2q+2}{2q+1}$ as needed.

6 Conclusion

We have given a complete analysis of deterministic semi-online algorithms for two related machines and non-increasing job sizes. It is left as an open problem to analyze the behavior of randomized algorithms for two machines. For a general setting of m machines, it should be difficult to give a complete analysis depending on the speeds. However, it is intriguing to close the open question: what is the best overall competitive ratio for m machines?

7 Acknowledgements

– We would like to thank Jiří Sgall, Joan Boyar and Gerhard Woeginger for helpful comments.
– We would like to thank an anonymous referee for thorough comments that significantly improved the presentation of the paper.
– A preliminary version of this paper appeared in Proc. of 27th Mathematical Foundations of Computer Science (MFCS'2002), pages 245-256.
– The research of Leah Epstein was supported in part by the Israel Science Foundation, (grant no. 250/01).
– The research of Lene M. Favrholdt was supported in part by the Danish Natural Science Research Council (SNF) and in part by the Future and Emerging Technologies program of the EU under contract number IST-1999-14186 (ALCOM-FT).

References

1. Y. Cho and S. Sahni. Bounds for List Schedules on Uniform Processors. *SIAM Journal on Computing*, 9(1):91–103, 1980.
2. G. Dobson. Scheduling Independent Tasks on Uniform Processors. *SIAM Journal on Computing*, 13(4):705–716, 1984.
3. L. Epstein, J. Noga, S. S. Seiden, J. Sgall, and G. J. Woeginger. Randomized Online Scheduling on Two Uniform Machines. *Journal of Scheduling*, 4(2):71–92, 2001.
4. D. K. Friesen. Tighter Bounds for LPT Scheduling on Uniform Processors. *SIAM Journal on Computing*, 16(3):554–560, 1987.

5. T. Gonzalez, O. H. Ibarra, and S. Sahni. Bounds for LPT Schedules on Uniform Processors. *SIAM Journal on Computing*, 6(1):155–166, 1977.
6. R. L. Graham. Bounds on Multiprocessing Timing Anomalies. *SIAM J. Appl. Math*, 17:416–429, 1969.
7. P. Mireault, J. B. Orlin, and R. V. Vohra. A Parametric Worst Case Analysis of the LPT Heuristic for Two Uniform Machines. *Operations Research*, 45:116–125, 1997.
8. S. Seiden, J. Sgall, and G. J. Woeginger. Semi-Online Scheduling with Decreasing Job Sizes. *Operations Research Letters*, 27(5):215–221, 2000.