Semi-online machine covering for two uniform machines

Xingyu Chen [*] Leah Epstein[†] Zhiyi Tan [‡]

Abstract

The machine covering problem deals with partitioning a sequence of jobs among a set of machines, so as to maximize the completion time of the least loaded machine. We study a semi-online variant, where jobs arrive one by one, sorted by non-increasing size. The jobs are to be processed by two uniformly related machines, with a speed ratio of $q \geq 1$. Each job has to be processed continuously, in a time slot dedicated to it on one of the machines. This assignment needs to be performed upon the arrival of the job. The length of the time slot, which is required for a specific job to run on a given machine, is equal to the size of the job divided by the speed of the machine. We give a complete competitive analysis of this problem by providing an algorithm of the best possible competitive ratio for every $q \geq 1$. We first give a tight analysis of the performance of a natural greedy algorithm LPT for the problem. To achieve the best possible performance for the semi-online problem, we use a combination of LPT, together with two alternative algorithms which we design. The new algorithms attain the best possible competitive ratios in the two intervals $q \in \left(1, \sqrt{1.5}\right)$ and $q \in \left(2.4856, 1+\sqrt{3}\right)$, respectively, whereas the greedy algorithm has the best possible competitive ratio for any other $q \geq 1$.

1 Introduction

In the machine covering problem [7, 6, 19, 2, 3, 14, 8, 18, 15, 5] (also called the Santa Claus problem [4, 1, 11]), n indivisible goods are to be partitioned among m clients. The goal is to distribute the goods in a way that the least satisfied client is still as pleased as possible. Each client i (where $1 \leq i \leq m$) values the goods using a non-negative vector $r_i = (r_i^1, r_i^2, \ldots, r_i^n)$. Let $J_i \subseteq \{1, 2, \ldots, n\}$ denote the subset of goods assigned to client i, such that $J_i \cap J_{i'} = \emptyset$ for any $i \neq i'$. The profit of a client i is $F_i = \sum\limits_{j \in J_i} r_i^j$. The objective is to maximize the minimum total profit of a client, that is, to maximize $\min\limits_{1 \leq i \leq m} F_i$. If the clients are *uniformly related*, then each of the goods can be assumed to have values p_j, and each client i has a parameter s_i, such that $r_i^j = \frac{p_j}{s_i}$ for any $1 \leq i \leq m$ and $1 \leq j \leq n$. This situation can occur if the goods

[*]Department of Mathematics, Zhejiang University, Hangzhou 310027, P. R. China. xingyuchan@gmail.com

[†]Department of Mathematics, University of Haifa, 31905 Haifa, Israel. lea@math.haifa.ac.il

[‡]Department of Mathematics, State Key Lab of CAD & CG, Zhejiang University, Hangzhou 310027, P. R. China. Supported by the National Natural Science Foundation of China (10671177, 60021201) and Zhejiang Provincial Natural Science Foundation of China (Y607079). tanzy@zju.edu.cn

have fixed monetary values. In this case, we have $F_i = \left(\sum_{j \in J_i} p_j\right)/s_i$. In this paper, we study the problem for the case of two clients. The problem is semi-online in the sense that goods arrive one by one, but they are sorted according to non-increasing values p_j. This type of study is common since the input is processed as a stream, and the required preprocessing can be performed efficiently.

We next define the problem using the terminology of scheduling. We study the semi-online variant of the machine covering problem on two uniformly related machines. The job sequence, denoted by $\{p_1, p_2, \ldots\}$, consists of independent jobs which arrive one by one, sorted by non-increasing size. We identify the jobs with their positive sizes and have $p_i \geq p_{i+1}$ for all $i \geq 1$. Let M_1 and M_2 denote two parallel, uniformly related machines, where the speed of M_i is s_i (for $i = 1, 2$), i.e., the time required for p_j to be executed on M_i is $\frac{p_j}{s_i}$ (for $j = 1, 2, \cdots n$ and $i = 1, 2$). We assume without loss of generality that $1 = s_1 \geq s_2 = \frac{1}{q}$, for some $q \geq 1$. If $q > 1$, M_1 is faster than M_2, and q is the speed ratio of the two machines. We call M_1 *the faster machine* and M_2 is called *the slower machine* (even if $q = 1$, where the machines are *identical*).

Jobs must be considered one by one, and each job is to be assigned without any additional information on further jobs. Nevertheless, the assignment takes place before time zero, and both jobs and machines are available at time zero. Furthermore, no preemption is allowed. The load of a machine is the total time required to complete all jobs assigned to it, i.e., if the set of jobs assigned to machine M_i is J_i then the load of M_i is $\left(\sum_{p_j \in J_i} p_j\right)/s_i$.

The objective value of an algorithm is the minimum load of any machine. The goal is to assign the jobs to the machines so as to maximize the objective value.

We measure an algorithm by its *competitive ratio*. Given an input job set I, let $C^A(I)$ (abbreviated by C^A, if the input I is clear from the context) and $C^*(I)$ (analogously abbreviated by C^*) be the objective values of the algorithm A and of an optimal schedule, respectively, of the input I. The competitive ratio of A is a function of the speed ratio q, which is denoted by $r^A(q)$. For every $q \geq 1$, $r^A(q)$ is defined to be the infimum $\mathcal{R}(q) \geq 1$ which satisfies $C^*(I) \leq \mathcal{R}(q)C^A(I)$ for any input sequence I, and a set of two machines with the speed ratio q.

A natural greedy algorithm for the problem is defined as follows.

Algorithm LPT**.** Assign a new job to the least loaded machine. In the case of a tie, i.e., if both machines have the same load, assign the current job to the faster machine.

Note that we see LPT as a semi-online algorithm, where the jobs arrive over list in a sorted order. This is equivalent to an offline variant where jobs are given as a set and at each time, the longest job is selected to be scheduled.

Intuitively, upon arrival of a new job, LPT tries to increase the minimum load. The choice of the faster machine in a case of a tie is not arbitrary. This machine requires a larger total size of jobs in order to have the same load as the slower machine. We call the assignment rule of LPT *the* LPT *rule*. Due to the LPT

rule, given a sequence of jobs with non-increasing sizes, the first two jobs are always assigned to different machines. Specifically, p_1 is always assigned to M_1 and p_2 is assigned to M_2. This last property is crucial in the case of large enough q, since in such cases, assigning the largest job to the slower machine immediately results in a large competitive ratio (see Section 6).

Note that another common variant of LPT for related machines assigns a job to the machine that would achieve a smaller load as a result of this assignment. We refer to this algorithm as *post-*LPT. This variant performs well for makespan minimization (minimization of the maximum load), while it performs poorly when the objective is maximization of the minimum load. In fact, in order to achieve a finite competitive ratio, an algorithm must assign the first two jobs to different machines, which is not always done by *post-*LPT.

Previous work. Online machine covering was previously studied for both identical machines and uniformly related machines. The offline problem is NP-hard (and strongly NP-hard for an arbitrary number of machines), but it admits a polynomial time approximation scheme (PTAS) [19, 10]. The best possible competitive ratio for the online problem with m identical machines is m (see [19]), and it is $q + 1$ for two uniformly related machines [8]. These competitive ratios are obtained by LPT.

Different approaches were applied in order to overcome these high competitive ratios. Such approaches were randomization (see [3], for the case of multiple identical machines), and assumptions on the input, that is, various semi-online models. Several papers considered semi-online variants for two uniformly related machines. In [2, 8], the variant where C^* is constant was investigated. The case where the total size of jobs is known in advance was studied in [18]. Luo, Sun and Huang [15], and in addition, Cao and Tan [5], considered the case where the size of the largest job is declared in advance.

The semi-online model studied in this paper, in which jobs arrive sorted by non-increasing size, was studied in the past for identical machines [7, 6] and for makespan minimization [16, 9].

Deuermeyer, Friesen and Langston [7] studied LPT, and showed an upper bound of $\frac{4}{3}$ on its competitive ratio. The tight ratio for this heuristic, $\frac{4m-2}{3m-1}$, was given by Csirik, Kellerer and Woeginger [6]. The above papers see the problem as an offline problem, and thus give only upper bounds, but it not difficult to see (using the examples of [6]) that for two and three machines, LPT is the best possible semi-online algorithm. This implies the competitive ratio 1.2 for $q = 1$, which is a special case of our results. For m uniformly related machines, a tight bound of m on the competitive ratio for the semi-online model was shown in [2].

As stated above, makespan minimization is the classical problem in which the goal is to minimize the maximum load of any machine. The semi-online variant with non-increasing job sizes and two machines was considered both for identical machines and related machines [13, 12, 17, 9, 16]. The upper bound for two identical machines follows from Graham [13]. Mireault, Orlin and Vohra [16] gave a complete analysis of *post-*LPT as a function of the speed ratio. Finally, a complete analysis of the best possible competitive ratio for two related machines was given in [9].

2 Main results

In this paper, we find the tight competitive ratio for semi-online machine covering with non-increasing job sizes.

We start with a complete analysis of LPT. We find the exact competitive ratio of LPT for all values of q and prove the following theorem in Section 4.

Theorem 2.1 *The exact competitive ratio of LPT is*

$$r^L(q) = \begin{cases} \frac{3q+3}{2q+3} & q \in [1, \sqrt{\frac{3}{2}} \approx 1.22474) \\ q & q \in [\sqrt{\frac{3}{2}}, \sqrt{2} \approx 1.41421) \\ \frac{2}{q} & q \in [\sqrt{2}, \frac{1+\sqrt{5}}{2} \approx 1.61803) \\ \frac{2q+2}{2q+1} & q \in [\frac{1+\sqrt{5}}{2}, \frac{1+\sqrt{7}}{2} \approx 1.82288) \\ \frac{2q+1}{q+2} & q \in [\frac{1+\sqrt{7}}{2}, \frac{1+\sqrt{13}}{2} \approx 2.30278) \\ \frac{3}{q} & q \in [\frac{1+\sqrt{13}}{2}, q_0) \\ \frac{q^2+q}{q^2+1} & q \in [q_0, 1+\sqrt{3} \approx 2.73205) \\ \frac{3q+2}{2q+3} & q \in [1+\sqrt{3}, 1+\sqrt{5} \approx 3.23607) \\ \frac{2q}{q+2} & q \in [1+\sqrt{5}, \infty), \end{cases}$$

where $q_0 \approx 2.4856$ is the largest real root of $q^3 - 2q^2 - 3 = 0$.

Many of the lower bound examples, which are used to show that the analysis of LPT is tight, can be converted into lower bounds for any semi-online algorithm (see Section 6). There exists however two intervals in which this is not the case. The reason for that becomes clear in Section 5, where two algorithms of smaller competitive ratios are designed for these specific cases. In fact, these algorithm achieve the best possible competitive ratio, as follows from the analysis in Section 5 and matching lower bounds which are proved in Section 6. Specifically, we prove the following theorem.

Theorem 2.2 *The optimal competitive ratio for semi-online scheduling on two related machines is*

$$r(q) = \begin{cases} \frac{6}{2q+3} & q \in [1, q_1) \\ \frac{2-q^2+\sqrt{q^4+4q^3+12q^2+16q+4}}{2(q+2)} & q \in [q_1, \frac{\sqrt{33}-1}{4} \approx 1.18614) \\ q & q \in [\frac{\sqrt{33}-1}{4}, \sqrt{2}) \\ \frac{2}{q} & q \in [\sqrt{2}, \frac{1+\sqrt{5}}{2}) \\ \frac{2q+2}{2q+1} & q \in [\frac{1+\sqrt{5}}{2}, \frac{1+\sqrt{7}}{2}) \\ \frac{2q+1}{q+2} & q \in [\frac{1+\sqrt{7}}{2}, \frac{1+\sqrt{13}}{2}) \\ \frac{3}{q} & q \in [\frac{1+\sqrt{13}}{2}, \frac{2+\sqrt{31}}{3} \approx 2.52259) \\ \frac{3q+2}{2q+3} & q \in [\frac{2+\sqrt{31}}{3}, 1+\sqrt{5}) \\ \frac{2q}{q+2} & q \in [1+\sqrt{5}, \infty), \end{cases}$$

where $q_1 \approx 1.0382$ *is the largest real root of* $4q^4 + 8q^3 + 15q^2 + 6q - 36 = 0$.

Comparing the two functions (see Figure 1), we can conclude that LPT is optimal for $q = 1$, $q \in [\sqrt{1.5}, q_0]$ and $q \in [1+\sqrt{3}, \infty)$. The total length of intervals where LPT is not optimal is approximately 0.471. Nevertheless, a careful design and analysis of alternative algorithms is required in order to achieve tight bounds for these cases. Note that both $r^L(q)$ and $r(q)$ attain their maximum value of 2 when $q \to \infty$. In other words, the overall competitive ratios of both $r^L(q)$ and $r(q)$ are 2, which is achieved for $q \to \infty$. Moreover,

$$r^L(q) \geq r(q) \geq \frac{2q}{q+2} \tag{1}$$

holds for any $q \geq 1$.

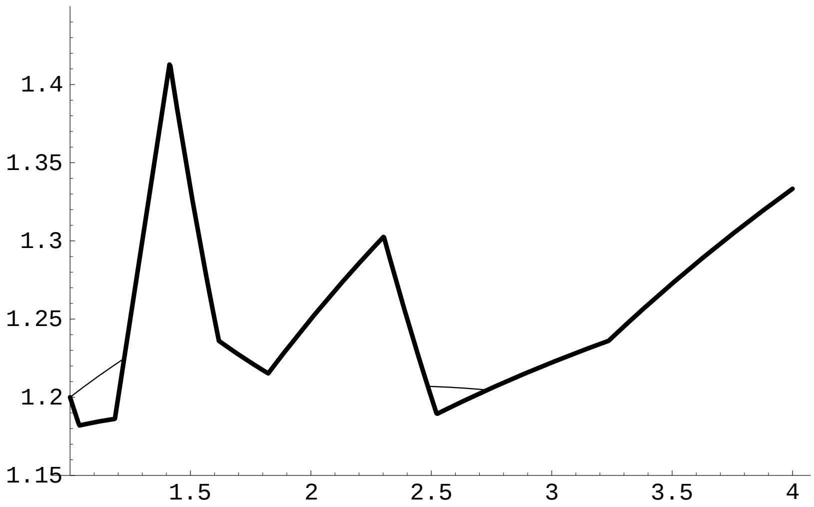

Figure 1: The competitive ratios of LPT and the optimal algorithm.

We next give some intuition for the partition into intervals. Both the behavior of LPT, and the semi-online problem in general, are dependent on the value of q. An attempt of performing a uniform analysis of LPT leads to proofs which do not hold for all values of q. Usually this simply means that the behavior of the competitive ratio changes at the infimum (or supremum) point, at which a proof no longer holds. From the point of view of lower bounds on the competitive ratio, a difficult example typically behaves differently starting from some point, and this point is often a breakpoint at which the competitive ratio function changes. In the cases where not every online algorithm can be forced into the same behavior as the one of LPT, we identified where LPT acts in a way which causes it to have a weaker performance than what is possible, and we define algorithms which behave similarly to LPT except for some special cases.

3 Preliminaries

In the next two sections, we prove the upper bounds on the competitive ratio in all cases by contradiction. We assume that $C^A < \frac{1}{r^A(q)} C^*$. We use T_i to denote the total size of jobs scheduled on M_i by Algorithm A, $i = 1, 2$. By scaling the instance we can assume that $C^* = 1$, and so $T_1 + T_2 \geq 1 + \frac{1}{q}$. For every value of q we consider a counter example which is minimal with respect to the number of jobs. We consider a specific optimal schedule to which we compare the performance of our algorithms.

We split out analysis into two situations according to the index of the machine which determines the objective value of the algorithm. We denote the job set containing the first j jobs by P_j. For each case, we analyze the potential structure of a minimal counter example. The following properties hold for any algorithm which assigns specific jobs according to the LPT rule (see below) and for any minimal counter example.

Situation $\mathcal{A}$. $C^A = \min\{T_1, qT_2\} = T_1 < \frac{1}{r^A(q)}$.

Since $T_1 < \frac{1}{r^A(q)} < 1$, we get $T_2 > 1 + \frac{1}{q} - \frac{1}{r^A(q)} > \frac{1}{q}$. Denote by p_l the last job assigned to M_2 by Algorithm A. Let L_i be the job set assigned to M_i just after p_l is assigned by the algorithm, $i = 1, 2$. Consequently, $P_l = L_1 \cup L_2$ and $l = |L_1| + |L_2|$. Let x_l be the total size of jobs which arrive after p_l, i.e., $x_l = T_1 + T_2 - \sum_{j=1}^{l} p_j$. These jobs are clearly assigned to M_1.

If p_l is assigned to M_2 according to the LPT rule, or more precisely, p_l is assigned to the machine with the smaller current load, then $T_1 \geq T_1 - x_l > q(T_2 - p_l)$. Hence

$$p_l > T_2 - \frac{T_1}{q} > \left(1 + \frac{1}{q} - \frac{1}{r^A(q)}\right) - \frac{1}{qr^A(q)} = (1 + \frac{1}{q})(1 - \frac{1}{r^A(q)}). \tag{2}$$

By (2), we can obtain upper bounds on $|L_1|$ and $|L_2|$. In fact, since

$$|L_1|(1 + \frac{1}{q})(1 - \frac{1}{r^A(q)}) < |L_1| p_l \leq T_1 < \frac{1}{r^A(q)},$$

we have

$$|L_1| < \frac{q}{(q+1)(r^A(q) - 1)}. \tag{3}$$

On the other hand,

$$\frac{1}{r^A(q)} > T_1 > q(T_2 - p_l) \geq q(|L_2| p_l - p_l) = q(|L_2| - 1) p_l > q(|L_2| - 1)(1 + \frac{1}{q})(1 - \frac{1}{r^A(q)}).$$

Therefore,

$$|L_2| < \frac{1}{(q+1)(r^A(q) - 1)} + 1. \tag{4}$$

Situation $\mathcal{B}$. $C^A = \min\{T_1, qT_2\} = qT_2 < \frac{1}{r^A(q)}$.

Since $T_2 < \frac{1}{qr^A(q)} < \frac{1}{q}$, $T_1 > 1 + \frac{1}{q} - \frac{1}{qr^A(q)} > 1$. Denote by p_u the last job assigned to M_1 in Algorithm A. Let U_i be the job set assigned to M_i just after p_u is assigned by the algorithm, $i = 1, 2$.

Consequently, $P_u = U_1 \cup U_2$ and $u = |U_1| + |U_2|$. Let x_u be the total size of jobs which arrive after p_u, i.e., $x_u = T_1 + T_2 - \sum_{j=1}^{u} p_j$.

We first show that in a minimal counter example we have $x_u = 0$. Consider an instance in which $x_u > 0$, thus the number of jobs in this instance is at least $u+1$. Consider the instance which contains only the jobs of P_u, and thus contains u jobs. The objective value of the algorithm is $q(T_2 - x_u)$. Consider the schedule obtained from an optimal schedule for the original input, where all jobs except for the jobs of P_u were removed. The objective value of this solution is at least $1 - q \cdot x_u > 1 - qT_2 > 0$, since the total size of jobs removed from each machine is at most x_u. We have $\frac{1-qx_u}{q(T_2-x_u)} \geq \frac{1}{qT_2} > r^A(q)$. Therefore, the modified input can serve as a smaller counter example.

If p_u is assigned to M_1 according to the LPT rule, we have $qT_2 \geq T_1 - p_u$. Then

$$p_u \geq T_1 - qT_2 > \left(1 + \frac{1}{q} - \frac{1}{qr^A(q)}\right) - q\frac{1}{qr^A(q)} = (1+\frac{1}{q})(1-\frac{1}{r^A(q)}). \tag{5}$$

Similarly to Situation $\mathcal{A}$, by (5) we have

$$(|U_1|-1)(1+\frac{1}{q})(1-\frac{1}{r^A(q)}) < (|U_1|-1)p_u = |U_1|p_u - p_u \leq T_1 - p_u \leq qT_2 < q\frac{1}{qr^A(q)}.$$

Hence,

$$|U_1| < \frac{q}{(q+1)(r^A(q)-1)} + 1. \tag{6}$$

On the other hand, since

$$|U_2|(1+\frac{1}{q})(1-\frac{1}{r^A(q)}) < |U_2|p_u \leq T_2 < \frac{1}{qr^A(q)},$$

we have

$$|U_2| < \frac{1}{(q+1)(r^A(q)-1)}. \tag{7}$$

Using these inequalities, we can find upper bounds on $|L_1|$ and $|L_2|$, if Situation $\mathcal{A}$ occurs, and otherwise on $|U_1|$ and $|U_2|$. These bounds must hold for a minimal counter example. The proof will exclude the existence of a minimal counter example and therefore of any counter example. This will be typically done by showing $C^* < 1$ (which contradicts our assumption, $C^* = 1$).

4 Analysis of LPT

In this section, we find the exact competitive ratio of LPT. We break the proof into several lemmas, each corresponding to a particular subset of intervals of q.

We first discuss several simple cases which may occur in the application of LPT. In Situation $\mathcal{A}$, if $|L_2| = 1$, then $p_l = p_2$ and $L_1 = \{p_1\}, L_2 = \{p_2\}$. If p_1 and p_2 are not assigned to M_1 together in the optimal schedule, then $C^* \le p_1 + x_l = T_1 < 1$. Otherwise,

$$C^* \le qx_l = q(T_1 - p_1) \le q(T_1 - p_2) = q(T_1 - T_2) < q\left(\frac{1}{r^L(q)} - \left(1 + \frac{1}{q} - \frac{1}{r^L(q)}\right)\right) \le 1,$$

where the last inequality is due to (1). In Situation $\mathcal{B}$, if $|U_1| = 1$ (or equivalently $|U_2| = 0$), then $p_u = p_1$ and $U_1 = \{p_1\}, U_2 = \emptyset$, which implies $p_1 > q(T_1 + T_2 - p_1)$. Clearly, LPT obtains an optimal schedule in this situation. So we assume $|L_2| \ge 2, |U_1| \ge 2$ and $|U_2| \ge 1$ in the following.

Lemma 4.1 *For $q \in [1, \sqrt{2})$, the competitive ratio of LPT is*

$$r^L(q) = \max\left\{\frac{3q+3}{2q+3}, q\right\} = \begin{cases} \frac{3q+3}{2q+3} & q \in [1, \sqrt{1.5}) \\ q & q \in [\sqrt{1.5}, \sqrt{2}). \end{cases}$$

Proof. We prove the upper bound first, and later show that it is tight.

By definition, if $1 \le q < \sqrt{2}$, then

$$\frac{1}{r^L(q)} = \min\left\{\frac{2q+3}{3q+3}, \frac{1}{q}\right\} \le \frac{1}{q}. \tag{8}$$

Situation $\mathcal{A}$. By the definition of $r^L(q)$ and (3), (4), we have $|L_1| \le 2, |L_2| \le 3$. We consider several cases according to the value of $|L_1|$ and $|L_2|$.

Case 1. $|L_1| = 1$ and $|L_2| = 2$.

Obviously, $L_1 = \{p_1\}$ and $L_2 = \{p_2, p_3\}$. By the pigeon-hole principle, any schedule must have a machine which processes at least two jobs of P_3, which holds for an optimal schedule as well. Thus, at most one job of P_3 is assigned to the other machine in the same schedule. Therefore, we have $C^* \le q(p_1 + x_l) = qT_1 < \frac{q}{r^L(q)} \le 1$ by (8), which leads to a contradiction.

Case 2. $|L_1| = 1$ and $|L_2| = 3$.

Obviously, $L_1 = \{p_1\}$ and $L_2 = \{p_2, p_3, p_4\}$. Consider all possible assignments of P_4 in the optimal schedule. If there exists a machine which processes at least three jobs of P_4, then we have $C^* \le q(p_1 + x_l) = qT_1 < 1$ by (8). Otherwise, both machines process two jobs of P_4. Recall that $q(p_2 + p_3) < p_1$ since p_4 is assigned to M_2 by LPT, we have $C^* \le q(p_2 + p_3 + x_l) < p_1 + qx_l \le q(p_1 + x_l) = qT_1 < 1$.

Case 3. $|L_1| = 2$ and $|L_2| = 2$.

Obviously, $L_1 = \{p_1, p_3\}$ and $L_2 = \{p_2, p_4\}$. Then $T_2 = p_2 + p_4 \le p_1 + p_3 \le T_1 < 1$. However, by (8), $T_2 > 1 + \frac{1}{q} - T_1 > 1 + \frac{1}{q} - \frac{1}{r^L(q)} \ge 1$, which is a contradiction.

Case 4. $|L_1| = 2$ and $|L_2| = 3$.

Note that when $\sqrt{1.5} \le q < \sqrt{2}$, $|L_2| < \frac{1}{(q+1)(q-1)} + 1 \le 3$ by (4). So we can assume $q < \sqrt{1.5}$ for this case. Then by (2), $p_l > \frac{1}{3}$.

If $L_1 = \{p_1, p_3\}$ and $L_2 = \{p_2, p_4, p_5\}$, then $p_1 \leq qp_2$ and $q(p_2 + p_4) \leq p_1 + p_3$. Together with $p_4 \geq p_5 > \frac{1}{3}$, we have

$$\begin{aligned} q(p_1 + p_2 + x_l) &\leq q(q+1)p_2 + qx_l \leq (q+1)(p_1 + p_3 - qp_4) + qx_l \\ &= (q+1)(T_1 - x_l - qp_4) + qx_l \leq (q+1)T_1 - q(q+1)p_4 \\ &< (q+1)\frac{2q+3}{3q+3} - \frac{q(q+1)}{3} \leq 1, \end{aligned} \tag{9}$$

where the last inequality holds for any $q \geq 1$. Otherwise, $L_1 = \{p_1, p_4\}$, $L_2 = \{p_2, p_3, p_5\}$, and thus $q(p_2 + p_3) \leq p_1 + p_4$. Together with $p_3 \geq p_4 \geq p_5 > \frac{1}{3}$, we get

$$\begin{aligned} q(p_1 + p_2 + x_l) &\leq qp_1 + (p_1 + p_4 - qp_3) + qx_l \leq (q+1)(p_1 + p_4 + x_l) - 2qp_4 \\ &= (q+1)T_1 - 2qp_4 < (q+1)\frac{2q+3}{3q+3} - \frac{2q}{3} = 1. \end{aligned} \tag{10}$$

Since there must exist a machine which processes at least three jobs of P_5 in the optimal schedule, at most two jobs of P_5 are assigned to the other machine. By (9) and (10), we have $C^* \leq q(p_1 + p_2 + x_l) < 1$, which is a contradiction.

Situation $\mathcal{B}$. By (6) and (7), we have $|U_1| \leq 3$ and $|U_2| \leq 2$. We consider several cases according to the value of $|U_1|$ and $|U_2|$.

Case 1. $|U_1| = 2$ and $|U_2| = 1$.

Obviously, $U_1 = \{p_1, p_3\}$, $U_2 = \{p_2\}$, and thus $p_1 \leq qp_2$. Since there must exist a machine which processes at least two jobs of P_3 in the optimal schedule, we have $C^* \leq qp_1 \leq q^2 p_2 = q^2 T_2 < 1$ by (8).

Case 2. $|U_1| = 2$ and $|U_2| = 2$.

Obviously, $U_1 = \{p_1, p_4\}$ and $U_2 = \{p_2, p_3\}$. We have $p_1 \leq q(p_2 + p_3)$ since p_4 is assigned to M_1. If there exists a machine which processes at least three jobs of P_4 in the optimal schedule, then $C^* \leq qp_1 \leq q^2(p_2 + p_3) = q^2 T_2 < 1$ as in Case 1. Otherwise, both machines process two jobs of P_4 in the optimal schedule. We also have $C^* \leq q(p_2 + p_3) = qT_2 < 1$.

Case 3. $|U_1| = 3$ and $|U_2| = 1$.

Obviously, $U_1 = \{p_1, p_3, p_4\}$, $U_2 = \{p_2\}$, and thus $p_1 + p_3 \leq qp_2$. Together with (8), we have $q(p_1) < q(p_1 + p_3) \leq q^2 p_2 = q^2 T_2 < 1$, and $q(p_2 + p_3) \leq q(p_1 + p_3) \leq q^2 p_2 = q^2 T_2 < 1$. As in Case 2, we get $C^* < 1$.

Case 4. $|U_1| = 3$ and $|U_2| = 2$.

Note that for $\sqrt{1.5} \leq q < \sqrt{2}$, $|U_2| < \frac{1}{(q+1)(q-1)} < 2$ by (7). So we can assume $q < \sqrt{1.5}$ for this case. Then by (5), $p_u > \frac{1}{3}$.

If $U_1 = \{p_1, p_3, p_5\}$ and $U_2 = \{p_2, p_4\}$, then $p_1 \leq qp_2$ since p_3 is assigned to M_1. Together with $p_4 \geq p_5 > \frac{1}{3}$, we have

$$q(p_1 + p_2) \leq q(qp_2 + p_2) = q(q+1)p_2 = q(q+1)(T_2 - p_4) < q(q+1)\left(\frac{2q+3}{q(3q+3)} - \frac{1}{3}\right) \leq 1,$$

for any $q \geq 1$. Otherwise, $U_1 = \{p_1, p_4, p_5\}$ and $U_2 = \{p_2, p_3\}$, then $p_1 + p_4 \leq q(p_2 + p_3)$ since p_5 is assigned to M_1. Together with $p_4 \geq p_5 > \frac{1}{3}$, we have

$$q(p_1 + p_2) \leq q(q(p_2 + p_3) - p_4 + p_2) \leq q((q+1)T_2 - p_4 - p_3)) < q(q+1)\frac{2q+3}{q(3q+3)} - \frac{2q}{3} = 1.$$

Since there must exist a machine which processes at least three jobs of P_5 in the optimal schedule, we get that $C^* \leq q(p_1 + p_2) < 1$.

Tight instances. If $q < \sqrt{1.5}$, then let the job sequence be $\{\frac{q+2}{3(q+1)}, \frac{q+3-q^2}{3q(q+1)}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$. To show that the sequence is non-increasing, note that $\frac{q+2}{3(q+1)} \geq \frac{q+3-q^2}{3q(q+1)}$ holds since this is equivalent to $2q^2 + q \geq 3$, and $\frac{q+3-q^2}{3q(q+1)} \geq \frac{1}{3}$ holds since it is equivalent to $q + 3 - q^2 \geq q^2 + q$, which holds for $q < \sqrt{1.5}$. If $q > 1$, LPT assigns the third job to M_2 since $\frac{q+2}{3(q+1)} > \frac{q+3-q^2}{3(q+1)}$. At this time, the loads are $\frac{q+2}{3(q+1)}$ (of M_1) and $\frac{2q+3}{3(q+1)}$ (of M_2). Assigning the next job to M_1 would result in equal loads of $\frac{2q+3}{3(q+1)}$. Since only one job remains at this time, we get $C^L = \frac{2q+3}{3q+3}$. In the optimal schedule, the jobs p_3, p_4 and p_5 are assigned to M_1 and the other jobs are assigned to M_2. Thus $C^* = 1$ and $\frac{C^*}{C^L} = \frac{3q+3}{2q+3}$. If $q = 1$ then the third job is assigned to M_1 and the fourth job to M_2, which gives the same result.

If $\sqrt{1.5} \leq q < \sqrt{2}$, then let the job sequence be $\{\frac{1}{q}, \frac{1}{q^2}, 1 - \frac{1}{q^2}\}$. The sequence is non-increasing for any $q \leq \sqrt{2}$. Clearly, LPT assigns p_1 to M_1 and p_2 to M_2, which results in equal loads of $\frac{1}{q}$. Since only one job is left at this time, $C^L = \frac{1}{q}$. In the optimal schedule, p_2, p_3 are assigned to M_1 and p_1 is assigned to M_2. Thus $C^* = 1$ and $\frac{C^*}{C^L} = q$. $\square$

Lemma 4.2 *For $q \in [\sqrt{2}, 1 + \sqrt{5})$, the competitive ratio of LPT is*

$$r^L(q) = \begin{cases} \frac{2}{q} & q \in [\sqrt{2}, \frac{1+\sqrt{5}}{2}) \\ \frac{2q+2}{2q+1} & q \in [\frac{1+\sqrt{5}}{2}, \frac{1+\sqrt{7}}{2}) \\ \frac{2q+1}{q+2} & q \in [\frac{1+\sqrt{7}}{2}, \frac{1+\sqrt{13}}{2}) \\ \frac{3}{q} & q \in [\frac{1+\sqrt{13}}{2}, q_0 \approx 2.4856) \\ \frac{q^2+q}{q^2+1} & q \in [q_0, 1 + \sqrt{3}) \\ \frac{3q+2}{2q+3} & q \in [1 + \sqrt{3}, 1 + \sqrt{5}). \end{cases}$$

Proof. It can be verified directly that

$$r^L(q) = \begin{cases} \max\{\frac{2}{q}, \frac{2q+2}{2q+1}, \frac{2q+1}{q+2}\} & q \in [\sqrt{2}, \frac{1+\sqrt{13}}{2}) \\ \max\{\frac{3}{q}, \frac{q^2+q}{q^2+1}, \frac{3q+2}{2q+3}\} & q \in [\frac{1+\sqrt{13}}{2}, 1 + \sqrt{5}) \end{cases} \tag{11}$$

and

$$r^L(q) \geq \max\left\{\frac{2}{q}, \frac{2q+2}{2q+1}, \frac{q^2+q}{q^2+1}, \frac{3q+2}{2q+3}\right\} \tag{12}$$

for all $q \in [\sqrt{2}, 1 + \sqrt{5})$.

Situation $\mathcal{A}$. By (3), (4) and simple algebraic calculation, we have $|L_1| \leq 3$ and $|L_2| \leq 2$.

Case 1. $|L_1| = 1$ and $|L_2| = 2$.

Obviously, $L_1 = \{p_1\}$, $L_2 = \{p_2, p_3\}$, and thus $qp_2 < p_1$. Consider all possible assignments of P_3 in the optimal schedule. If p_1 is the only job of P_3 which is assigned to M_1, then it is trivial that $C^* \leq p_1 + x_l = T_1 < 1$. If p_1 is the only job of P_3 which is assigned to M_2, then by (12),

$$C^* \leq p_2 + p_3 + x_l < \frac{2}{q}p_1 + x_l \leq \max\left\{\frac{2}{q}, 1\right\}(p_1 + x_l) = \max\left\{\frac{2}{q}, 1\right\}T_1 < \max\left\{\frac{2}{q}, 1\right\}\frac{1}{r^L(q)} \leq 1,$$

since $p_3 \leq p_2 < \frac{p_1}{q}$.

If p_1 is assigned together with at least one other job of P_3, then

$$\begin{aligned} C^* &\leq q(p_2 + x_l) = q(p_2 + T_1 - p_1) \leq -q(q-1)p_2 + qT_1 \leq -\frac{q(q-1)}{2}(p_2 + p_3) + qT_1 \\ &= -\frac{q(q-1)}{2}T_2 + qT_1 < -\frac{q(q-1)}{2}\left(1 + \frac{1}{q} - \frac{1}{r^L(q)}\right) + \frac{q}{r^L(q)} \leq 1, \end{aligned}$$

where the last inequality is equivalent to $r^L(q) \geq \frac{q^2+q}{q^2+1}$, which is valid due to (12).

Case 2. $|L_1| = 2$ and $|L_2| = 2$.

Obviously, $L_1 = \{p_1, p_3\}$, $L_2 = \{p_2, p_4\}$, and thus $qp_2 < p_1 + p_3$. Consider all possible assignments of P_4 in the optimal schedule. If there are at least two jobs of P_4 assigned to M_2, then at most two jobs of P_4 are assigned to M_1. We obtain by (2) and (12),

$$\begin{aligned} C^* &\leq p_1 + p_2 + x_l < p_1 + \frac{1}{q}(p_1 + p_3) + x_l \leq (1 + \frac{1}{q})(p_1 + p_3 + x_l) - p_3 = (1 + \frac{1}{q})T_1 - p_3 \\ &< \left(1 + \frac{1}{q}\right)\frac{1}{r^L(q)} - \left(1 + \frac{1}{q}\right)\left(1 - \frac{1}{r^L(q)}\right) \leq 1, \end{aligned}$$

where the last inequality is equivalent to $r^L(q) \geq \frac{2q+2}{2q+1}$, which is valid due to (12). If there is at most one job of P_4 assigned to M_2 and p_1 is assigned to M_1, then

$$\begin{aligned} C^* &\leq q(p_2 + x_l) = q(p_2 + T_1 - p_1 - p_3) \leq -q(q-1)p_2 + qT_1 \leq -\frac{q(q-1)}{2}(p_2 + p_4) + qT_1 \\ &= -\frac{q(q-1)}{2}T_2 + qT_1 < -\frac{q(q-1)}{2}\left(1 + \frac{1}{q} - \frac{1}{r^L(q)}\right) + \frac{q}{r^L(q)} \leq 1 \end{aligned}$$

as in Case 1. Otherwise, the only job of P_4 which is assigned to M_2 is p_1. By (11), we also have

$$C^* \leq q(p_1 + x_l) = q(T_1 - p_3) < \frac{q}{r^L(q)} - q\left(1 + \frac{1}{q}\right)\left(1 - \frac{1}{r^L(q)}\right) \leq 1$$

for $\sqrt{2} \leq q < \frac{1+\sqrt{13}}{2}$, since the last expression is equivalent to $r^L(q) \geq \frac{2q+1}{q+2}$, and

$$\begin{aligned} C^* &\leq p_2 + p_3 + p_4 + x_l \leq 3p_2 + x_l < \frac{3}{q}(p_1 + p_3) + x_l \\ &\leq \max\left\{\frac{3}{q}, 1\right\}(p_1 + p_3 + x_l) = \max\left\{\frac{3}{q}, 1\right\}T_1 < \max\left\{\frac{3}{q}, 1\right\}\frac{1}{r^L(q)} \leq 1 \end{aligned}$$

when $\frac{1+\sqrt{13}}{2} \le q < 1+\sqrt{5}$.

Case 3. $|L_1| = 3$ and $|L_2| = 2$.

Obviously, $L_1 = \{p_1, p_3, p_4\}$, $L_2 = \{p_2, p_5\}$, and thus $qp_2 < p_1 + p_3 + p_4$. Consider all possible assignments of P_5 in the optimal schedule. If there is at most one job of P_5 assigned to M_2, then

$$C^* \le q(p_1 + x_l) = q(T_1 - p_3 - p_4) \le qT_1 - 2qp_5 < \frac{q}{r^L(q)} - 2q\left(1+\frac{1}{q}\right)\left(1-\frac{1}{r^L(q)}\right) \le 1,$$

where the last inequality is equivalent to $r^L(q) \ge \frac{3q+2}{2q+3}$, which is valid due to (12). Otherwise, at least two jobs of P_5 are assigned to M_2. Since at most three jobs of P_5 are assigned to M_1, we have

$$\begin{aligned} C^* &\le p_1 + p_2 + p_3 + x_l < p_1 + \frac{p_1 + p_3 + p_4}{q} + p_3 + x_l \le (1+\frac{1}{q})T_1 - p_4 \\ &< \left(1+\frac{1}{q}\right)\frac{1}{r^L(q)} - \left(1+\frac{1}{q}\right)\left(1-\frac{1}{r^L(q)}\right) \le 1 \end{aligned}$$

as in Case 2.

Situation $\mathcal{B}$. By (6) and (7), we have $|U_1| \le 4$ and $|U_2| \le 1$.

Case 1. $|U_1| = 2$ and $|U_2| = 1$.

Obviously, $U_1 = \{p_1, p_3\}$, $U_2 = \{p_2\}$, and thus $p_1 \le qp_2$. Consider all possible assignments of P_3 in the optimal schedule. If p_1 is the only job of P_3 which is assigned to M_1, then $C^* \le p_1 \le qp_2 = qT_2 < 1$. If p_1 is the only job of P_3 which is assigned to M_2, then by (12), $C^* \le p_2 + p_3 \le 2p_2 = 2T_2 < \frac{2}{qr^L(q)} \le 1$. If p_1 is assigned together with at least one other job of P_3, then $C^* \le qp_2 = qT_2 < 1$.

Case 2. $|U_1| = 3$ and $|U_2| = 1$.

Obviously, $U_1 = \{p_1, p_3, p_4\}$, $U_2 = \{p_2\}$, and thus $p_1 + p_3 \le qp_2$. Consider all possible assignments of P_4 in the optimal schedule. If there are at least two jobs of P_4 assigned to M_2, we obtain

$$\begin{aligned} C^* &\le p_1 + p_2 \le qp_2 - p_3 + p_2 \le (1+q)p_2 - p_3 \\ &= (1+q)T_2 - p_3 < (1+q)\frac{1}{qr^L(q)} - \left(1+\frac{1}{q}\right)\left(1-\frac{1}{r^L(q)}\right) \le 1\,, \end{aligned}$$

as in previous cases. If there is at most one job of P_4 assigned to M_2 and p_1 is assigned to M_1, it is trivial that $C^* \le qp_2 = qT_2 < 1$. Otherwise, the only job of P_4 which is assigned to M_2 is p_1. By (5) and (11), we also have

$$C^* \le qp_1 \le q(qp_2 - p_3) \le q(qT_2 - p_3) < \frac{q}{r^L(q)} - q\left(1+\frac{1}{q}\right)\left(1-\frac{1}{r^L(q)}\right) \le 1$$

for $\sqrt{2} \le q < \frac{1+\sqrt{13}}{2}$, and

$$C^* \le p_2 + p_3 + p_4 \le 3p_2 \le 3p_2 = 3T_2 < \frac{3}{qr^L(q)} \le 1$$

for $\frac{1+\sqrt{13}}{2} \le q < 1+\sqrt{5}$.

Case 3. $|U_1| = 4$ and $|U_2| = 1$.

Obviously $U_1 = \{p_1, p_3, p_4, p_5\}$, $U_2 = \{p_2\}$, and thus $p_1 + p_3 + p_4 \le qp_2$. Consider all possible assignments of P_5 in the optimal schedule. If there is at most one job of P_5 assigned to M_2 in the optimal schedule, then by (5) and (12), we have

$$C^* \le qp_1 \le q(qp_2 - p_3 - p_4) \le q(qT_2 - 2p_4) \le \frac{q}{r^L(q)} - 2q\left(1 + \frac{1}{q}\right)\left(1 - \frac{1}{r^L(q)}\right) \le 1.$$

Otherwise, at least two jobs of P_5 are assigned to M_2. By (5) and (12), we also have

$$\begin{aligned} C^* &\le p_1 + p_2 + p_3 \le qp_2 - p_4 + p_2 \le (1+q)p_2 - p_4 = (1+q)T_2 - p_4 \\ &< (1+q)\frac{1}{qr^L(q)} - \left(1 + \frac{1}{q}\right)\left(1 - \frac{1}{r^L(q)}\right) \le 1. \end{aligned}$$

Tight instances If $\sqrt{2} \le q < \frac{1+\sqrt{5}}{2}$, then let the job sequence be $\{\frac{1}{q}, \frac{1}{2}, \frac{1}{2}\}$. After the assignment of the first two jobs, the loads of M_1 and M_2 (respectively) are $\frac{1}{q}$ and $\frac{q}{2} \ge \frac{1}{q}$, which holds for $q \ge \sqrt{2}$. Therefore, LPT assigns p_1, p_3 to M_1 and p_2 to M_2, which results in $C^L = \frac{q}{2}$. In the optimal schedule, p_2, p_3 are assigned to M_1 and p_1 is assigned to M_2. Thus $C^* = 1$ and $\frac{C^*}{C^L} = \frac{2}{q}$.

If $\frac{1+\sqrt{5}}{2} \le q < \frac{1+\sqrt{7}}{2}$, then let the job sequence be $\{\frac{2q^2-1}{2q(q+1)}, \frac{2q+1}{2q(q+1)}, \frac{1}{2q}, \frac{1}{2q}\}$. The sequence is non-increasing since $2q^2 - 1 \ge 2q + 1$ is equivalent to $q \ge \frac{1+\sqrt{5}}{2}$. Since $2q^2 - 1 < 2q^2 + q$, LPT assigns p_3 to M_1. At this time, the loads are both equal to $\frac{2q+1}{2(q+1)}$. Since only one job is left at this time, we have $C^L = \frac{2q+1}{2q+2}$. In the optimal schedule, p_1, p_2 are assigned to M_1 and p_3, p_4 are assigned to M_2. Thus $C^* = 1$ and $\frac{C^*}{C^L} = \frac{2q+2}{2q+1}$.

If $\frac{1+\sqrt{7}}{2} \le q < \frac{1+\sqrt{13}}{2}$, then let the job sequence be $\{\frac{1}{q}, \frac{q+2}{q(2q+1)}, \frac{q^2-1}{q(2q+1)}, \frac{q^2-1}{q(2q+1)}\}$. To show that the sequence is non-increasing, we need to show $2q + 1 \ge q + 2 \ge q^2 - 1$, which holds for $1 \le q \le \frac{1+\sqrt{13}}{2}$. At the time when the first two jobs were assigned, the load of M_2 is larger than the load of M_1 ($\frac{q^2+2q}{q(2q+1)}$ versus $\frac{2q+1}{2q(q+1)}$). LPT assigns the next job to M_1 which results in equal loads of $\frac{q+2}{2q+1}$. At this time, only one job is left and thus $C^L = \frac{q+2}{2q+1}$. In the optimal schedule, p_2, p_3, p_4 are assigned to M_1 and p_1 is assigned to M_2. Thus $C^* = 1$ and $\frac{C^*}{C^L} = \frac{2q+1}{q+2}$.

If $\frac{1+\sqrt{13}}{2} \le q < q_0$, then let the job sequence be $\{\frac{1}{q}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$. We claim that LPT assigns p_1, p_3, p_4 to M_1 and p_2 to M_2. After two jobs are assigned, the load of M_1 is $\frac{1}{q}$, while the load of M_2 is $\frac{q}{3} > \frac{1}{q}$ for $q > \sqrt{3}$, thus the third job is assigned to M_1. After the assignment of two jobs to M_1, its load becomes $\frac{1}{q} + \frac{1}{3}$, while the load of M_2 is $\frac{q}{3}$. Since for $q \ge \frac{1+\sqrt{13}}{2}$, $\frac{1}{q} + \frac{1}{3} \le \frac{q}{3}$, an additional job is assigned to M_1, which results in $C^L = \frac{q}{3}$. In the optimal schedule, p_2, p_3, p_4 are assigned to M_1 and p_1 is assigned to M_2. Thus $C^* = 1$ and $\frac{C^*}{C^L} = \frac{3}{q}$.

If $q_0 \le q < 1 + \sqrt{3}$, then let the job sequence be $\{\frac{q}{q+1} + \varepsilon, \frac{1}{q+1} - \varepsilon, \frac{1}{q+1} - \varepsilon, \frac{1}{q(q+1)} + \varepsilon\}$, where $\varepsilon > 0$ is a small enough real number. Clearly, LPT assigns p_3 to M_2, which results in $C^L \le p_1 + p_4 = \frac{q^2+1}{q^2+q} + 2\varepsilon$. In the optimal schedule, p_1, p_2 are assigned to M_1 and p_3, p_4 are assigned to M_2. Thus $C^* = 1$ and $\frac{C^*}{C^L} \to \frac{q^2+q}{q^2+1}$ (letting ε tend to 0).

If $1+\sqrt{3} \le q < 1+\sqrt{5}$, then let the job sequence be $\{\frac{1}{q}, \frac{2q+3}{q(3q+2)}, \frac{q^2-1}{q(3q+2)}, \frac{q^2-1}{q(3q+2)}, \frac{q^2-1}{q(3q+2)}\}$. The sequence is non-increasing if $3q+2 \ge 2q+3 \ge q^2-1$, which holds for $1 \le q \le 1+\sqrt{5}$.

After the assignment of the first two jobs, the load of M_2 is $\frac{2q^2+3q}{q(3q+2)}$ and the load of M_1 is $\frac{3q+2}{q(3q+2)}$, so the next job is assigned to M_1. This results in the load $\frac{q^2+3q+1}{q(3q+2)}$, thus the fourth job is assigned to M_1 as well, which results in equal loads of $\frac{2q+3}{3q+2}$. At this time, a single job is left, thus $C^L = \frac{2q+3}{3q+2}$. In the optimal schedule, p_2, p_3, p_4, p_5 are assigned to M_1 and p_1 is assigned to M_2. Thus $C^* = 1$ and $\frac{C^*}{C^L} = \frac{3q+2}{2q+3}$. □

Lemma 4.3 *For $q \in [1+\sqrt{5}, \infty)$, the competitive ratio of LPT is $r^L(q) = \frac{2q}{q+2}$.*

Proof. By (4) and (7), we have $|L_2| < 2$ if Situation $\mathcal{A}$ occurs and $|U_2| = 0$, if Situation $\mathcal{B}$ occurs. Therefore, the upper bound follows from the discussion before.

A tight instance. Let the job sequence be $\{\frac{1}{2}, \frac{1}{2}, \frac{1}{q}\}$. Clearly, LPT assigns p_1, p_3 to M_1 and p_2 to M_2, which results in $C^L = \frac{q+2}{2q}$. In the optimal schedule, p_1, p_2 are assigned to M_1 and p_3 is assigned to M_2. Thus $C^* = 1$ and $\frac{C^*}{C^L} = \frac{2q}{q+2}$. □

5 Two new algorithms

In this section, we introduce two new algorithms, and analyze their competitive ratios. In the next section we prove matching lower bounds. In particular, we show that the competitive ratios are smaller than those of LPT, and thus LPT is not optimal in the intervals discussed here.

The goal of these algorithms is to behave differently from LPT in the cases where LPT clearly makes an incorrect choice. As we saw in the previous section, the most difficult cases to deal with are the first few jobs. After many jobs have been assigned, LPT becomes a reasonable strategy for all cases. Thus we need to reconsider the assignment rule of the first few jobs.

For small values of q, it is unclear whether assigning the first job to the faster machine is always the correct thing to do. Our algorithm $LM1$ always makes the opposite choice. The next job must be assigned to the faster machine, in order to avoid an unbounded competitive ratio. The assignment of the third job depends on the exact sizes. An additional interval in which LPT does not achieve the best possible competitive ratio is treated in a similar way. Due to the large value of q, it is impossible to switch places of the first two jobs, but the third and fourth jobs must be assigned very carefully.

Algorithm $LM1$

1. Assign p_1 to M_2, and p_2 to M_1.
2. If $p_1 \ge \frac{r(q)}{q} p_2$, assign p_3 to M_1, otherwise assign p_3 to M_2.
3. Assign the remaining jobs according to the LPT rule.

Lemma 5.1 *For $q \in [1, \sqrt{1.5})$, the competitive ratio of $LM1$ is*

$$\begin{aligned} r(q) &= \max\left\{\frac{6}{2q+3}, \frac{2-q^2+\sqrt{q^4+4q^3+12q^2+16q+4}}{2(q+2)}, q\right\} \\ &= \begin{cases} \frac{6}{2q+3} & q \in [1, q_1) \\ \frac{2-q^2+\sqrt{q^4+4q^3+12q^2+16q+4}}{2(q+2)} & q \in [q_1, \frac{\sqrt{33}-1}{4}) \\ q & q \in [\frac{\sqrt{33}-1}{4}, \sqrt{1.5}). \end{cases} \end{aligned}$$

Claim 5.1 *For any $1 \le q \le \sqrt{1.5}$, $(1+\frac{q}{r(q)})(\frac{1}{r(q)} - q(1+\frac{1}{q})(1-\frac{1}{r(q)})) \le 1$ and $r(q) \le \frac{q}{q-1}$.*

Proof. If $q \le \sqrt{1.5}$, we have $r(q) \le r^L(q) = \frac{3q+3}{2q+3} < \frac{q}{q-1}$. Let $r'(q) = \frac{2-q^2+\sqrt{q^4+4q^3+12q^2+16q+4}}{2(q+2)}$. In fact, $r'(q)$ is the positive solution of

$$\left(1+\frac{q}{r(q)}\right)\left(\frac{1}{r(q)} - q\left(1+\frac{1}{q}\right)\left(1-\frac{1}{r(q)}\right)\right) = 1 \tag{13}$$

with respect to $r(q)$. Since $\frac{q+2}{r(q)} - q - 1 > 0$, and $r(q) \ge r'(q)$, we have

$$\begin{aligned} (1+\frac{q}{r(q)})(\frac{1}{r(q)} - q(1+\frac{1}{q})(1-\frac{1}{r(q)})) &= (1+\frac{q}{r(q)})(\frac{q+2}{r(q)} - q - 1) \le (1+\frac{q}{r'(q)})(\frac{q+2}{r'(q)} - q - 1) \\ &= (1+\frac{q}{r'(q)})(\frac{1}{r'(q)} - q(1+\frac{1}{q})(1-\frac{1}{r'(q)})) = 1. \end{aligned}$$

□

In the proof of the competitive ratio, we use the following technical lemma.

Lemma 5.2 *Let T_i^* be the total size of jobs scheduled on M_i in the optimal schedule, for $i = 1, 2$. Since $C^* = \min\{T_1^*, qT_2^*\}$, for any $a, b > 0$, we have*

$$C^* \le \frac{aT_1^* + bqT_2^*}{a+b}. \tag{14}$$

Proof. Since $C^* \le T_1^*$ and $C^* \le qT_2^*$, and $a, b > 0$, we get $(a+b)C^* \le aT_1^* + bqT_2^*$. □

In Situation $\mathcal{A}$, we denote by δ_i^* the total size of jobs which arrive after p_l, and are scheduled on M_i in the optimal schedule, $i = 1, 2$. Then $\delta_1^* + \delta_2^* = x_l$, and for any $a, b > 0$, we have $a\delta_1^* + bq\delta_2^* \le \max\{a, bq\}x_l$. Note that we do not use a similar definition for Situation $\mathcal{B}$ since we consider a minimal counter example, and thus we assume $x_u = 0$.

Proof. (Proof of Lemma 5.1).

Situation $\mathcal{A}$. $C^{LM1} = \min\{T_1, qT_2\} = T_1 < \frac{1}{r(q)}$.

We have $T_1 < \frac{1}{r(q)} < 1$, $T_2 > 1 + \frac{1}{q} - \frac{1}{r(q)} > \frac{1}{q}$. If $|L_2| = 1$, then $p_l = p_1$. No matter which machine p_1 is assigned to in the optimal schedule, we have $C^* \leq qx_l = qT_1 < \frac{q}{r(q)} \leq 1$. So we assume $|L_2| \geq 2$ in the following, and thus $l \geq 3$.

Case 1. $p_1 \geq \frac{r(q)}{q} p_2$.

According to Algorithm $LM1$, p_3 is assigned to M_1. Thus p_l must be assigned to M_2 due to the LPT rule, and $|L_1| \geq 2$. By the definition of $r(q)$ and (3),(4), we have $|L_1| \leq 2$ and $|L_2| \leq 3$. Hence, $|L_1| = 2$ and $|L_2| = 2$ or 3. We consider several subcases according to the value of $|L_2|$.

Case 1.1 $|L_2| = 2$.

Obviously, $L_1 = \{p_2, p_3\}$ and $L_2 = \{p_1, p_4\}$. Consider all possible assignments of P_4 in the optimal schedule. If there exists a machine which processes at least three jobs of P_4, recall that $qp_1 < p_2 + p_3$ since p_4 is assigned to M_2 by the LPT rule, we have

$$C^* \leq q(p_1 + x_l) < p_2 + p_3 + qx_l \leq q(p_2 + p_3 + x_l) = qT_1 < 1.$$

Otherwise, both machines process two jobs of P_4, we have $C^* \leq q(p_2 + p_3 + x_l) = qT_1 < 1$.

Case 1.2 $|L_2| = 3$.

Obviously, $L_1 = \{p_2, p_3\}$ and $L_2 = \{p_1, p_4, p_5\}$, and thus $q(p_1 + p_4) < p_2 + p_3$. Since there must exist a machine which processes at most two jobs of P_5 in the optimal schedule, by (2) and Claim 5.1, we have

$$\begin{aligned} C^* &\leq q(p_1 + p_2 + x_l) \leq q\left(p_1 + \frac{q}{r(q)} p_1 + x_l\right) \leq \left(1 + \frac{q}{r(q)}\right)(qp_1 + x_l) \\ &< \left(1 + \frac{q}{r(q)}\right)(p_2 + p_3 - qp_4 + x_l) \leq \left(1 + \frac{q}{r(q)}\right)(T_1 - qp_5) \\ &< \left(1 + \frac{q}{r(q)}\right)\left(\frac{1}{r(q)} - q\left(1 + \frac{1}{q}\right)\left(1 - \frac{1}{r(q)}\right)\right) \leq 1. \end{aligned}$$

Case 2. $p_1 < \frac{r(q)}{q} p_2$.

According to Algorithm $LM1$, p_3 is assigned to M_2. If $|L_2| = 2$, then $L_1 = \{p_2\}$ and $L_2 = \{p_1, p_3\}$. Since there must exist a machine which processes at least two jobs of P_3 in the optimal schedule, we have

$$C^* \leq q(p_1 + x_l) < r(q)p_2 + qx_l \leq r(q)(p_2 + x_l) = r(q)T_1 < 1.$$

Otherwise $|L_2| \geq 3$, and then p_l is assigned to M_2 by LPT rule, and thus $|L_1| \leq 2$ by the definition of $r(q)$ and (3). However, p_4 must be assigned to M_1 and $p_2 + p_4 \leq p_1 + p_3 \leq q(p_1 + p_3)$ implies that at least one additional job must be assigned to M_1 before p_l is assigned to M_2. Therefore $|L_1| \geq 3$, which is a contradiction.

Situation $\mathcal{B}$. $C^{LM1} = \min\{T_1, qT_2\} = qT_2 < \frac{1}{r(q)}$.

We have $T_2 < \frac{1}{qr(q)} < \frac{1}{q}$, $T_1 > 1 + \frac{1}{q} - \frac{1}{qr(q)} > 1$. Since $p_2 \in U_1, p_1 \in U_2$ and $p_2 \leq p_1 \leq T_2 \leq qT_2$, we obtain $|U_1| \geq 2$.

Case 1. $p_1 \geq \frac{r(q)}{q} p_2$.

According to Algorithm $LM1$, p_3 is assigned to M_1. If $|U_1| = 2$, obviously $U_1 = \{p_2, p_3\}$ and $U_2 = \{p_1\}$. Since there must exist a machine which processes at least two jobs of P_3 in the optimal schedule, we have $C^* \leq qp_1 = qT_2 < 1$. So we suppose $|U_1| \geq 3$, and p_u, where $u \geq 4$, must be assigned by the LPT rule. By (6) and (7), we have $|U_1| \leq 3$ and $|U_2| \leq 2$. Hence $|U_1| = 3$.

Case 1.1 $|U_2| = 1$.

Obviously, $U_1 = \{p_2, p_3, p_4\}$ and $U_2 = \{p_1\}$. Thus $p_2 + p_3 \leq qp_1$. Consider all possible assignments of P_4 in the optimal schedule. If there exists a machine which processes at least three jobs of P_4, then we have $C^* \leq qp_1 = qT_2 < 1$. Otherwise,

$$C^* \leq q(p_2 + p_3) \leq q^2 p_1 = q^2 T_2 < q^2 \frac{1}{qr(q)} \leq 1.$$

Case 1.2 $|U_2| = 2$.

Obviously, $U_1 = \{p_2, p_3, p_5\}$ and $U_2 = \{p_1, p_4\}$. Since there must exist a machine which processes at most two jobs of P_5 in the optimal schedule, by (5) and (13), we have

$$\begin{aligned} C^* &\leq q(p_1 + p_2) \leq q \left(p_1 + \frac{q}{r(q)} p_1 \right) \leq \left(1 + \frac{q}{r(q)} \right) (q(p_1 + p_4) - qp_4) \\ &\leq \left(1 + \frac{q}{r(q)} \right) (qT_2 - qp_5) < \left(1 + \frac{q}{r(q)} \right) \left(\frac{1}{r(q)} - q \left(1 + \frac{1}{q} \right) \left(1 - \frac{1}{r(q)} \right) \right) \leq 1. \end{aligned}$$

Case 2. $p_1 < \frac{r(q)}{q} p_2$.

According to Algorithm $LM1$, p_3 is assigned to M_2. Note that $p_2 + p_4 \leq q(p_1 + p_3)$, p_4 is assigned to M_1 and since M_2 must be less loaded after the assignment of p_u, then $u \geq 5$, and $|U_1| \geq 3$. On the other hand, since p_u is assigned to M_1 by LPT rule, we have $|U_1| \leq 3$ by (6). Hence, $U_1 = \{p_2, p_4, p_5\}$ and $U_2 = \{p_1, p_3\}$. Consider all possible assignments of P_5 in the optimal schedule. Recall that there must exist a machine which processes at most two jobs of P_5. If these jobs are not the pair p_1 and p_2, that is, this is a different pair, or a single job, then we have $C^* \leq q(p_1 + p_3) = qT_2 < 1$. If p_1, p_2 are assigned to M_1, by Lemma 5.2 and (14), with $a = 3q, b = 2$, we have $C^* \leq \frac{3qT_1^* + 2qT_2^*}{3q+2}$. We use $3qT_1^* + 2qT_2^* = 3q(p_1 + p_2) + 2q(p_3 + p_4 + p_5) \leq 6qp_1 + 6qp_3 = 6qT_2 < \frac{6q}{r(q)}$, and get $C^* < \frac{6}{(3q+2)r(q)} \leq \frac{6}{(2q+3)r(q)} \leq 1$, where the last inequality is due to the definition of $r(q)$, and the previous one is due to $q \geq 1$.

Otherwise, if p_1, p_2 are assigned to M_2, we take $a = 2q, b = 3$, and get $C^* \leq \frac{2qT_1^* + 3qT_2^*}{2q+3}$. In this case, $2qT_1^* + 3qT_2^* = 2q(p_3 + p_4 + p_5) + 3q(p_1 + p_2) \leq 6qp_1 + 6qp_3 = 6qT_2 < \frac{6q}{r(q)}$. This gives $C^* < \frac{6}{(2q+3)r(q)} \leq 1$. □

Algorithm $LM2$

1. Assign p_1 to M_1, and p_2 to M_2.

2. If $p_1 < qr(q)p_2$, assign p_3 to M_1, otherwise assign p_3 to M_2.

3. Denote by T_i^s the total size of jobs scheduled on M_i before p_4 is scheduled, $i = 1, 2$. If $T_1^s < 2T_2^s + p_4$, assign p_4 to M_1, otherwise assign p_4 to M_2.

4. Assign the remaining jobs according to the LPT rule.

Lemma 5.3 *For $q \in [q_0, 1+\sqrt{3})$, the competitive ratio of $LM2$ is*

$$r(q) = \max\left\{\frac{3}{q}, \frac{3q+2}{2q+3}\right\} = \begin{cases} \frac{3}{q} & q \in [q_0, \frac{2+\sqrt{31}}{3}) \\ \frac{3q+2}{2q+3} & q \in [\frac{2+\sqrt{31}}{3}, 1+\sqrt{3}). \end{cases}$$

Proof. It can be verified directly that

$$r(q) \geq \frac{2q+2}{2q+1} \tag{15}$$

for $q \in [q_0, \frac{2+\sqrt{31}}{3})$.

Situation $\mathcal{A}$. $C^{LM2} = \min\{T_1, qT_2\} = T_1 < \frac{1}{r(q)}$.

We have $T_1 < \frac{1}{r(q)} < 1$, $T_2 > 1 + \frac{1}{q} - \frac{1}{r(q)} > \frac{1}{q}$. If $|L_2| = 1$, then $p_l = p_2$ and $L_1 = \{p_1\}, L_2 = \{p_2\}$. Consider all possible assignments of P_2 in the optimal schedule. If p_1, p_2 are assigned to the same machine, then

$$C^* \leq qx_l = q(T_1 - p_1) \leq q(T_1 - p_2) = q(T_1 - T_2) < q\left(\frac{1}{r(q)} - \left(1 + \frac{1}{q} - \frac{1}{r(q)}\right)\right) \leq 1$$

by (1). Otherwise $C^* \leq p_1 + x_l = T_1 < 1$. So we assume $|L_2| \geq 2$ in the following.

Case 1. $|L_2| = 2$.

If $p_1 \geq qr(q)p_2$, then p_3 is assigned to M_2. Since $|L_2| = 2$, we have $L_1 = \{p_1\}$ and $L_2 = \{p_2, p_3\}$. Consider all possible assignments of P_3 in the optimal schedule. If p_1 is assigned to M_2, then by $r(q) \geq \frac{3}{q}$,

$$C^* \leq p_2 + p_3 + x_l < qr(q)p_2 + x_l \leq p_1 + x_l = T_1 < 1.$$

If p_1 is the only job of P_3 which is assigned to M_1, it is trivial that $C^* \leq p_1 + x_l = T_1 < 1$. Otherwise, we have

$$\begin{aligned} C^* &\leq q(p_2 + x_l) = q(p_2 + T_1 - p_1) \leq q(p_2 + T_1 - qr(q)p_2) \leq qT_1 - q(qr(q) - 1)\left(\frac{p_2 + p_3}{2}\right) \\ &= qT_1 - \frac{q(qr(q)-1)}{2}T_2 < \frac{q}{r(q)} - \frac{q(qr(q)-1)}{2}\left(1 + \frac{1}{q} - \frac{1}{r(q)}\right) \leq 1, \end{aligned}$$

where the last inequality is equivalent to $(q^2+q)r(q)^2 - (q^2+q-1)r(q) - q \geq 0$, which is valid due to the following: $(q^2+q)r(q)^2 - (q^2+q-1)r(q) - q = (q^2+q)r(q)(r(q)-1) + r(q) - q \geq 3(q+1)(r(q)-1) + r(q) - q = (3q+4)r(q) - (4q+3)$, by $r(q) \geq \frac{3}{q}$. Since $r(q) \geq \frac{3q+2}{2q+3} \geq \frac{4q+3}{3q+4}$ for any $q \geq 1$, the property follows.

Now we consider the case $p_1 < qr(q)p_2$. Thus p_3 is assigned to M_1, and $T_1^s = p_1 + p_3, T_2^s = p_2$.

Case 1.1 $p_1 + p_3 < 2p_2 + p_4$.

In this case, p_4 is assigned to M_1 and p_l is assigned to M_2 due to the LPT rule, since $l \geq 5$. By the definition of $r(q)$ and (3), we have $|L_1| \leq 3$. Hence, $|L_1| = 3$ and $L_1 = \{p_1, p_3, p_4\}, L_2 = \{p_2, p_5\}$.

Consider all possible assignments of P_5 in the optimal schedule. If there exists a machine which processes at least four jobs in P_5, then by (2),

$$\begin{aligned} C^* &\leq q(p_1 + x_l) = q(T_1 - p_3 - p_4) \leq q(T_1 - 2p_5) \\ &< q\left(\frac{1}{r(q)} - 2\left(1 + \frac{1}{q}\right)\left(1 - \frac{1}{r(q)}\right)\right) \leq \frac{3q+2}{r(q)} - (2q+2) \leq 1, \end{aligned}$$

where the last inequality is equivalent to $r(q) \geq \frac{3q+2}{2q+3}$. Otherwise, by (2), (15) and $qp_2 \leq p_1 + p_3 + p_4$ since p_5 is assigned to M_2, we have

$$\begin{aligned} C^* &\leq p_1 + p_2 + p_3 + x_l < p_1 + \frac{p_1 + p_3 + p_4}{q} + p_3 + x_l \leq \left(1 + \frac{1}{q}\right)(p_1 + p_3 + p_4 + x_l) - p_4 \\ &\leq \left(1 + \frac{1}{q}\right) T_1 - p_5 < \left(1 + \frac{1}{q}\right) \frac{1}{r(q)} - \left(1 + \frac{1}{q}\right)\left(1 - \frac{1}{r(q)}\right) \leq 1. \end{aligned}$$

Case 1.2 $p_1 + p_3 \geq 2p_2 + p_4$.

According to the definition of Algorithm $LM2$, p_4 is assigned to M_2. Obviously, $L_1 = \{p_1, p_3\}$ and $L_2 = \{p_2, p_4\}$. Consider all possible assignments of P_4 in the optimal schedule. Firstly, suppose p_1 is assigned to M_2. Then

$$C^* \leq p_2 + p_3 + p_4 + x_l \leq 2p_2 + p_4 + x_l \leq p_1 + p_3 + x_l = T_1 \leq 1.$$

Secondly, suppose p_1 is assigned to M_1 with at least two other jobs of P_4. Then by (1),

$$\begin{aligned} C^* &\leq q(p_2 + x_l) = q(p_2 + T_1 - (p_1 + p_3)) \leq q(p_2 + T_1 - (2p_2 + p_4)) = q(T_1 - p_2 - p_4)) \\ &= q(T_1 - T_2) < q\left(\frac{1}{r(q)} - 1 - \frac{1}{q} + \frac{1}{r(q)}\right) \leq 1. \end{aligned}$$

Thirdly, if p_1 is the only job of P_4 assigned to M_1, or it is assigned to M_1 with p_3 or with p_4, then $C^* \leq p_1 + p_3 + x_l = T_1 < 1$. Finally, suppose p_1, p_2 are assigned to M_1. By (14) with $a = 4q, b = 3$ and $r(q) \geq \frac{3q+2}{2q+3} \geq \frac{6q}{4q+3}$ when $q \leq 6$, we have

$$\begin{aligned} C^* &\leq \frac{4qT_1^* + 3qT_2^*}{4q+3} = \frac{4q(p_1 + p_2 + \delta_1^*) + 3q(p_3 + p_4 + \delta_2^*)}{4q+3} \\ &\leq \frac{4qp_1 + 2q(2p_2 + p_4) + 3qp_3 + qp_4 + (4q\delta_1^* + 3q\delta_2^*)}{4q+3} \\ &\leq \frac{4qp_1 + 2q(p_1 + p_3) + 4qp_3 + (4q\delta_1^* + 3q\delta_2^*)}{4q+3} \\ &= \frac{6q(p_1 + p_3 + x_l)}{4q+3} < \frac{6qT_1}{4q+3} \leq \frac{6q}{4q+3} \cdot \frac{1}{r(q)} \leq 1. \end{aligned}$$

Case 2. $|L_2| \geq 3$.

If $\{p_3, p_4\} \nsubseteq L_2$ or $|L_2| \geq 4$, then p_l is assigned to M_2 due to the LPT rule, since $l \geq 5$. By the definition of $r(q)$ and (4), we have $|L_2| \leq 2$, which is a contradiction. Hence, $\{p_3, p_4\} \subseteq L_2$ and $|L_2| \leq 3$. In other words, $L_2 = \{p_2, p_3, p_4\}$ and $L_1 = \{p_1\}$. According to Algorithm $LM2$, we have $p_1 \geq qr(q)p_2$ and $p_1 \geq 2(p_2 + p_3) + p_4$.

Consider all possible assignments of P_4 in the optimal schedule. If there exists a machine which processes a single job of P_4, which is p_1, then $C^* \leq \max\{p_1 + x_l, p_2 + p_3 + p_4 + x_l\} = p_1 + x_l = T_1 < 1$. Otherwise, by (1),

$$C^* \leq q(p_2 + p_3 + x_l) \leq q(p_1 - p_2 - p_3 - p_4 + x_l) = q(T_1 - T_2) < q\left(\frac{1}{r(q)} - 1 - \frac{1}{q} + \frac{1}{r(q)}\right) \leq 1.$$

Situation $\mathcal{B}$. $C^{LM2} = \min\{T_1, qT_2\} = qT_2 < \frac{1}{r(q)}$.

We have $T_2 < \frac{1}{qr(q)} < \frac{1}{q}, T_1 > 1 + \frac{1}{q} - \frac{1}{qr(q)} > 1$. If $|U_1| = 1$, then $U_1 = \{p_1\}, U_2 = \emptyset$. Since $x_u = 0$, in this case $C^* = 0$. Thus, we assume $|U_1| \geq 2$ in the following.

Case 1. $|U_1| = 2$.

Case 1.1 If $p_1 < qr(q)p_2$, then p_3 is assigned to M_1 according to Algorithm $LM2$. Obviously, $U_1 = \{p_1, p_3\}$ and $U_2 = \{p_2\}$. Consider all possible assignments of P_3 in the optimal schedule. If p_1 is assigned to M_1, then $C^* \leq \max\{p_1, qp_2\} < qr(q)p_2 = qr(q)T_2 < 1$. Otherwise $C^* \leq p_2 + p_3 \leq 2p_2 \leq qp_2 = qT_2 < 1$.

Next we consider the option where $p_1 \geq qr(q)p_2$. According to Algorithm $LM2$, p_3 is assigned to M_2, and thus $T_1^s = p_1$, $T_2^s = p_2 + p_3$.

Case 1.2 $p_1 < 2(p_2 + p_3) + p_4$.

According to Algorithm $LM2$, p_4 is assigned to M_1. Obviously, $U_1 = \{p_1, p_4\}$ and $U_2 = \{p_2, p_3\}$. Consider all possible assignments of P_4 in the optimal schedule. If there exists a machine which processes only the job p_1 in P_4, then by $r(q) \geq \frac{3}{q}$,

$$\begin{aligned} C^* &\leq \max\{p_1, p_2 + p_3 + p_4\} \leq \max\{p_1, 3p_2\} \leq \max\{p_1, qr(q)p_2\} \\ &= p_1 < 2(p_2 + p_3) + p_4 \leq 3(p_2 + p_3) = 3T_2 \leq \frac{3}{qr(q)} \leq 1. \end{aligned}$$

Otherwise, $C^* \leq q(p_2 + p_3) = qT_2 < 1$.

Case 1.3 $p_1 \geq 2(p_2 + p_3) + p_4$.

According to Algorithm $LM2$, p_4 is assigned to M_2. Then p_u is assigned to M_1 by the LPT rule, since $u \geq 5$. By (7), we have $|U_2| \leq 1$, which is a contradiction.

Case 2. $|U_1| \geq 3$.

If $|U_2| \geq 2$, then p_u is assigned to M_1 due to the LPT rule, since $u \geq 5$. By (7), we have $|U_2| \leq 1$, which is a contradiction. So $|U_2| = 1$ and $U_1 \supseteq \{p_1, p_3, p_4\}, U_2 = \{p_2\}$. According to the algorithm, we have $p_1 < qr(q)p_2$ and $p_1 + p_3 < 2p_2 + p_4$.

If $|U_1| = 3$, then $U_1 = \{p_1, p_3, p_4\}$. Consider all possible assignments of P_4 in the optimal schedule. If M_2 processes exactly one job of P_4, then using $q < 3$,

$$C^* \leq \max\{p_2 + p_3 + p_4, qp_2\} \leq 3p_2 = 3T_2 < \frac{3}{qr(q)} \leq 1.$$

Otherwise, by $r(q) \geq \frac{3}{q}$,

$$C^* \leq p_1 + p_2 < 2p_2 + p_4 - p_3 + p_2 \leq 3p_2 = 3T_2 < \frac{3}{qr(q)} \leq 1.$$

If $|U_1| = 4$, then $U_1 = \{p_1, p_3, p_4, p_5\}$. We have $p_1 + p_3 + p_4 \leq qp_2$ by the LPT rule. Consider all possible assignments of P_5 in the optimal schedule. If there exists a machine which processes at least four jobs in P_5, then by (5), and by $r(q) \geq \frac{3q+2}{2q+3}$, we get

$$\begin{aligned} C^* &\leq qp_1 \leq q(qp_2 - p_3 - p_4) \leq q(qp_2 - 2p_5) = q^2 p_2 - 2qp_5 \\ &\leq q^2 T_2 - 2q\left(1 + \frac{1}{q}\right)\left(1 - \frac{1}{r(q)}\right) < \frac{q}{r(q)} - 2(q+1)\left(1 - \frac{1}{r(q)}\right) \leq 1. \end{aligned} \tag{16}$$

If two jobs are assigned to M_1 in the optimal schedule, then by $r(q) \geq \frac{3}{q}$,

$$C^* \leq p_1 + p_2 < 2p_2 + p_4 - p_3 + p_2 \leq 3p_2 = 3T_2 < \frac{3}{qr(q)} \leq 1.$$

If M_1 processes three jobs, and M_2 processes two jobs (in the optimal schedule), we have $2qT_1^* + qT_2^* \leq 2q(p_1 + p_2 + p_3) + q(p_4 + p_5)$. By (14) with $a = 2q, b = 1$ and (15), we have

$$\begin{aligned} C^* &\leq \frac{2qT_1^* + qT_2^*}{2q+1} \leq \frac{2q(p_1 + p_2 + p_3) + q(p_4 + p_5)}{2q+1} \leq \frac{2q(qp_2 - p_4 + p_2) + q \cdot 2p_4}{2q+1} \\ &\leq \frac{2q(q+1)p_2}{2q+1} = \frac{2q(q+1)T_2}{2q+1} < \frac{2q+2}{(2q+1)r(q)} < 1. \end{aligned}$$

By the definition of $r(q)$ and (6), if $|U_1| \geq 4$, then p_u is assigned by the LPT rule and therefore $|U_1| \leq 4$. The proof is thus completed. □

6 Lower bounds

In this section, we present valid job sequences (i.e., sequence sorted by non-increasing size) which allow us to prove lower bounds which match the upper bounds from the previous sections. All sequences have at most five jobs. Let r_s be the ratio of objective values of the optimal schedule and a schedule given by an arbitrary algorithm A just after p_s is assigned, $s \geq 1$. Obviously, $\frac{C^*}{C^A} \geq r_s$ for any $s \geq 1$.

Given a job sequence, if p_1, p_2 are assigned to the same machine, then $r_2 \to \infty$. So we only need to consider algorithms that assign the first two jobs to different machines in the following.

Furthermore, for $q \geq \frac{\sqrt{33}-1}{4}$ we have $r(q) \leq q$. Therefore, in all cases except for the first two intervals, we assume that the first job is assigned to M_1. If this is not the case, then a second (and last) job of size $p_2 = \frac{p_1}{q}$ arrives. To avoid an unbounded competitive ratio, this job must be assigned to M_1. We get $C^* = p_1$ whereas $C^A = \frac{p_1}{q}$, thus $r_2 = q$.

Interval 1. $q \in [1, q_1)$, $r(q) = \frac{6}{2q+3}$.

The sequence consists of five jobs of sizes $\{\frac{1}{2q}, \frac{1}{2q}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$. If p_3, p_4 are assigned to the same machine, then after four jobs, $C^* \geq \frac{1}{2q} + \frac{1}{3} = \frac{2q+3}{6q}$, and the algorithm has a machine with a single job of size $\frac{1}{2q}$, so $C^A \leq \frac{1}{2}$, therefore, $r_4 \geq \frac{2q+3}{3q} > r(q)$. Otherwise, an optimal schedule assigns the last three jobs to M_1 and $C^* = 1$. The algorithm has a machine with just one job of size $\frac{1}{3}$, so $r_5 \geq \frac{1}{q(\frac{2q+3}{6q})} = \frac{6}{2q+3} = r(q)$.

Interval 2. $q \in [q_1, \frac{\sqrt{33}-1}{4})$, $r(q) = \frac{2-q^2+\sqrt{q^4+4q^3+12q^2+16q+4}}{2(q+2)}$.

It can be verified directly by the definition of $r(q)$ that $q + r(q) > r(q)^3 > r(q)^2$ for $q \in [q_1, \frac{\sqrt{33}-1}{4})$, which will be used frequently in the following.

If p_1 is assigned to M_1, the sequence consists of five jobs of sizes

$$\left\{ \frac{r(q)}{q}, \frac{(q+3-q^2)r(q)}{q^2(q+2)}, \frac{(q+1)r(q)}{q(q+2)}, \frac{(q+1)r(q)}{q(q+2)}, \frac{(q+1)r(q)}{q(q+2)} \right\}.$$

The sequence is of sizes non-increasing since $q^2 + 2q \geq q + 3 - q^2 \geq q^2 + q$ for any $1 \leq q \leq \sqrt{1.5}$. If p_3 and p_4 are assigned to the same machine, then

$$r_4 \geq \frac{\min\{p_1 + p_4, q(p_2 + p_3)\}}{\max\{p_1, qp_2\}} = \frac{\frac{(2q+3)r(q)}{q(q+2)}}{\frac{r(q)}{q}} = \frac{2q+3}{q+2} > \frac{3q+3}{2q+3} > r(q).$$

Otherwise

$$r_5 \geq \frac{\min\{p_3 + p_4 + p_5, q(p_1 + p_2)\}}{\max\{p_1 + p_4, q(p_2 + p_3)\}} = \frac{\frac{3(q+1)r(q)}{q(q+2)}}{\frac{(2q+3)r(q)}{q(q+2)}} = \frac{3q+3}{2q+3} > r(q).$$

If p_1 is assigned to M_2, the sequence consists of five jobs of sizes

$$\left\{ \frac{r(q)}{q}, 1, \frac{q}{r(q)}, \frac{q + r(q) - r(q)^2}{qr(q)}, \frac{q + r(q) - r(q)^2}{qr(q)} \right\}.$$

The sequence of sizes is non-increasing since $r(q) \geq q > 1$ in this interval. If p_3 is assigned to M_2, then

$$r_3 \geq \frac{\min\{p_2 + p_3, qp_1\}}{p_2} = \min\left\{ \frac{q + r(q)}{r(q)}, r(q) \right\} = r(q).$$

Thus we only need to consider algorithms that assign p_3 to M_1. In this case, if p_4 is assigned to M_1, then

$$r_4 \geq \frac{\min\{p_2 + p_3, q(p_1 + p_4)\}}{\min\{p_2 + p_3 + p_4, qp_1\}} = \frac{\frac{q+r(q)}{r(q)}}{\min\left\{ \frac{(q+1)r(q)+q(q+1)-r(q)^2}{qr(q)}, r(q) \right\}} = \frac{\frac{q+r(q)}{r(q)}}{r(q)} = \frac{q + r(q)}{r(q)^2} > r(q).$$

Otherwise, by (13),

$$r_5 \geq \frac{\min\{p_3+p_4+p_5, q(p_1+p_2)\}}{\max\{p_2+p_3, q(p_1+p_4)\}} = \frac{\min\left\{\frac{q^2+2q+2r(q)-2r(q)^2}{qr(q)}, q+r(q)\right\}}{\frac{q+r(q)}{r(q)}} = \frac{q+r(q)}{\frac{q+r(q)}{r(q)}} = r(q).$$

Recall the in the remaining intervals, we only need to consider algorithms that assign p_1 to M_1 and p_2 to M_2.

Interval 3. $q \in [\frac{\sqrt{33}-1}{4}, \sqrt{2})$, $r(q) = q$.

The sequence consists of three jobs of sizes $\{\frac{1}{q}, \frac{1}{q^2}, 1-\frac{1}{q^2}\}$.

The sequence is non-increasing since $q^2 \leq 2$. After the first two jobs are assigned, the machines have equal loads. Thus, we get $r_3 \geq \frac{1}{\frac{1}{q}} = q = r(q)$.

In the remaining intervals, the full instances are similar to those shown in Section 4. Therefore, we have already shown that they are non-increasing (in the cases where this is not immediately seen from the sequence).

Interval 4. $q \in [\sqrt{2}, \frac{1+\sqrt{5}}{2})$, $r(q) = \frac{2}{q}$.

The sequence consists of three jobs of sizes $\{\frac{1}{q}, \frac{1}{2}, \frac{1}{2}\}$. The loads after the first two jobs are assigned are $\frac{1}{q}$ and $\frac{q}{2}$, so $C^A \leq \frac{q}{2}$ and $C^* = 1$. We get $r_3 \geq \frac{qp_1}{qp_2} = r(q)$.

Interval 5. $q \in [\frac{1+\sqrt{5}}{2}, \frac{1+\sqrt{7}}{2})$, $r(q) = \frac{2q+2}{2q+1}$.

The sequence consists of four jobs of sizes $\{\frac{2q^2-1}{2q(q+1)}, \frac{2q+1}{2q(q+1)}, \frac{1}{2q}, \frac{1}{2q}\}$.

For the prefix of three jobs we have $C^* = \frac{2q^2+q}{2q(q+1)}$, since an optimal schedule assigns p_1 and p_3 to M_1, and p_2 to M_2. If p_3 is assigned to M_2, then $C^A = \frac{2q^2-1}{2q(q+1)}$, so $r_3 = \frac{q(2q+1)}{2q^2-1} > r(q)$ for any $q \geq 1$. Otherwise the machines have an equal load after this assignment, so $r_4 \geq r(q)$.

Interval 6. $q \in [\frac{1+\sqrt{7}}{2}, \frac{1+\sqrt{13}}{2})$, $r(q) = \frac{2q+1}{q+2}$.

The sequence consists of four jobs of sizes $\{\frac{1}{q}, \frac{q+2}{q(2q+1)}, \frac{q^2-1}{q(2q+1)}, \frac{q^2-1}{q(2q+1)}\}$.

For the prefix of three jobs $C^* = \frac{q+2}{2q+1}$, since an optimal schedule assigns p_1 and p_3 to M_1, and p_2 to M_2. If p_3 is assigned to M_2, then $C^A \leq \frac{1}{q}$, so $r_3 \geq \frac{q(q+2)}{2q+1} > r(q)$ for any $q \geq 1$. Otherwise the loads after three jobs are assigned are $\frac{q+2}{2q+1}$, so $r_4 \geq r(q)$.

Interval 7. $q \in [\frac{1+\sqrt{13}}{2}, \frac{2+\sqrt{31}}{3})$, $r(q) = \frac{3}{q}$.

The sequence consists of four jobs of sizes $\{\frac{1}{q}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$.

For the prefix of three jobs $C^* = \frac{q+3}{3q}$, since an optimal schedule assigns p_1 and p_3 to M_1, and p_2 to M_2, and $\frac{q}{3} \geq \frac{q+3}{3q}$ for $q \geq \frac{1+\sqrt{13}}{2}$. If p_3 is assigned to M_2, then $C^A \leq \frac{1}{q}$, so $r_3 \geq \frac{q+3}{3} > r(q)$ for any $q > 2$. Otherwise $r_4 \geq r(q)$, since the larger load after three jobs is $\frac{q}{3}$.

Interval 8. $q \in [\frac{2+\sqrt{31}}{3}, 1+\sqrt{5})$, $r(q) = \frac{3q+2}{2q+3}$.

The sequence consists of five jobs of sizes $\{\frac{1}{q}, \frac{2q+3}{q(3q+2)}, \frac{q^2-1}{q(3q+2)}, \frac{q^2-1}{q(3q+2)}, \frac{q^2-1}{q(3q+2)}\}$.

For the prefix of three jobs $C^* = \frac{q+3}{3q}$, since an optimal schedule can assign p_1 and p_2 to M_1, and p_3 to M_2, if p_3 is assigned to M_2 by the algorithm, then $C^A \leq \frac{1}{q}$, so $r_3 \geq \frac{\min\{p_1+p_2, qp_3\}}{\frac{1}{q}} = \min\{\frac{5q+5}{3q+2}, \frac{q(q^2-1)}{3q+2}\} > r(q)$, for any $q > 2.5$.

Otherwise, if p_3 is assigned to M_1, we consider the prefix of four jobs. For this prefix we have $C^* = \frac{2q+3}{3q+2}$, since an optimal schedule assigns p_1, p_3 and p_4 to M_1, and p_2 to M_2. If p_4 is assigned to M_2, then $C^A \leq \frac{q^2+3q+1}{q(3q+2)}$, so $r_4 \geq \frac{q(2q+3)}{q^2+3q+1} > r(q)$. for any $q \geq 1$.

Finally, if p_4 is assigned to M_1, the loads of both machines after four jobs have been assigned are $\frac{2q+3}{3q+2}$, therefore $r_5 \geq r(q)$.

Interval 9. $q \in [1+\sqrt{5}, \infty)$, $r(q) = \frac{2q}{q+2}$.

The sequence consists of three jobs of sizes $\{\frac{1}{2}, \frac{1}{2}, \frac{1}{q}\}$. Obviously, the best that the algorithm can do is to assign p_3 to M_1. We get $r_3 \geq \frac{1}{p_1+p_3} = r(q)$.

References

[1] A. Asadpour and A. Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. In *Proc. of the 39th Annual ACM Symposium on Theory of Computing (STOC2007)*, pages 114–121, 2007.

[2] Y. Azar and L. Epstein. On-line machine covering. In *Proc. of the 5th Annual European Symposium on Algorithms (ESA'97)*, pages 23–36, 1997.

[3] Y. Azar and L. Epstein. On-line machine covering. *Journal of Scheduling*, 1(2):67–77, 1998.

[4] N. Bansal and M. Sviridenko. The santa claus problem. In *38th ACM Symposium on Theory of Computing (STOC2006)*, pages 31–40, 2006.

[5] S. Cao and Z. Tan. Online uniform machine covering with the known largest size. *Progress in Natural Science*, 17:1271–1278, 2007.

[6] J. Csirik, H. Kellerer, and G. Woeginger. The exact LPT-bound for maximizing the minimum completion time. *Operations Research Letters*, 11:281–287, 1992.

[7] B. Deuermeyer, D. Friesen, and M. Langston. Scheduling to maximize the minimum processor finish time in a multiprocessor system. *SIAM Journal on Discrete Methods*, 3:190–196, 1982.

[8] L. Epstein. Tight bounds for online bandwidth allocation on two links. *Discrete Applied Math.*, 148(2):181–188, 2005.

[9] L. Epstein and L. M. Favrholdt. Optimal non-preemptive semi-online scheduling on two related machines. *J. Algorithms*, 57(1):49–73, 2005.

[10] L. Epstein and J. Sgall. Approximation schemes for scheduling on uniformly related and identical parallel machines. *Algorithmica*, 39(1):43–57, 2004.

[11] U. Feige. On allocations that maximize fairness. In *Proc. of the 19th annual ACM-SIAM symposium on Discrete algorithms (SODA2008)*, pages 287–293, 2008.

[12] T. Gonzalez, O. H. Ibarra, and S. Sahni. Bounds for LPT schedules on uniform processors. *SIAM Journal on Computing*, 6(1):155–166, 1977.

[13] R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM J. Appl. Math*, 17:416–429, 1969.

[14] Y. He and Y. Jiang. Optimal semi-online preemptive algorithms for machine covering on two uniform machines. *Theoretical Computer Science*, 339(2-3):293–314, 2005.

[15] R. Luo, S. Sun, and W. Huang. Semi on-line scheduling problem for maximizing the minimum machine completion time on two uniform machines. *Journal of Systems Science and Complexity*, 19, 2006.

[16] 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.

[17] S. Seiden, J. Sgall, and G. Woeginger. Semi-online scheduling with decreasing job sizes. *Operations Research Letters*, 27(5):215–221, 2000.

[18] Z. Y. Tan and S. J. Cao. Semi-online machine covering on two uniform machines with known total size. *Computing*, 78(4):369–378, 2006.

[19] G. Woeginger. A polynomial time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters*, 20:149–154, 1997.