

Angewandte Mathematik und Optimierung Schriftenreihe
Applied Mathematics and Optimization Series
AMOS # 27(2015)

György Dósa, Armin Fügenschuh, Zhiyi Tan, Zsolt Tuza,
and Krzysztof Węsek

Semi-Online Scheduling on Two Uniform
Machines with Known Optimum
Part I: Tight Lower Bounds

Herausgegeben von der
Professur für Angewandte Mathematik
Professor Dr. rer. nat. Armin Fügenschuh

Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg
Fachbereich Maschinenbau
Holstenhofweg 85
D-22043 Hamburg

Telefon: +49 (0)40 6541 3540
Fax: +49 (0)40 6541 3672

e-mail: appliedmath@hsu-hh.de
URL: <http://www.hsu-hh.de/am>

Angewandte Mathematik und Optimierung Schriftenreihe (AMOS), ISSN-Print 2199-1928
Angewandte Mathematik und Optimierung Schriftenreihe (AMOS), ISSN-Internet 2199-1936

Semi-Online Scheduling on Two Uniform Machines with Known Optimum, Part I: Tight Lower Bounds

György Dósa ^{*} Armin Fügenschuh [†] Zhiyi Tan [‡]
Zsolt Tuza ^{§¶} Krzysztof Węsek ^{**}

Abstract

This problem is about to schedule a number of jobs of different lengths on two uniform machines with given speeds 1 and $s \geq 1$, so that the overall completion time, i.e., the makespan, is earliest possible. We consider a semi-online variant (introduced for equal speeds) by Azar and Regev, where the jobs arrive one after the other, while the scheduling algorithm knows the optimum value of the corresponding offline problem.

One can ask how close any possible algorithm could get to the optimum value, that is, to give a lower bound on the competitive ratio: the supremum over ratios between the value of the solution given by the algorithm and the optimal offline solution. We contribute to this question by constructing tight lower bounds for all values of s in the intervals $[\frac{1+\sqrt{21}}{4}, \frac{3+\sqrt{73}}{8}] \approx [1.3956, 1.443]$ and $[\frac{5}{3}, \frac{4+\sqrt{133}}{9}] \approx [\frac{5}{3}, 1.7258]$, except a very narrow interval, approximately $[1.6934, 1.6963]$, where our new lower bound is “almost tight”.

A novel feature of the (rather complicated) construction of malicious input sequences is that our method goes several levels deeper than the earlier ones in the literature.

Keywords: Semi-online scheduling, makespan minimization, machine scheduling, lower bound.

^{*}Department of Mathematics, University of Pannonia, Veszprém, Hungary, dosagy@almos.vein.hu, **corresponding author**

[†]Helmut Schmidt University / University of the Federal Armed Forces Hamburg, Holstenhofweg 85, 22043 Hamburg, Germany, fuegenschuh@hsu-hh.de

[‡]Department of Mathematics, Zhejiang University, Hangzhou, Peoples Republic of China, tanzhy@zju.edu.cn

[§]Department of Computer Science and Systems Technology, University of Pannonia, Veszprém, Hungary, tuza@dcs.uni-pannon.hu

[¶]Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, Budapest, Hungary

^{||}Helmut Schmidt University / University of the Federal Armed Forces Hamburg, Holstenhofweg 85, 22043 Hamburg, Germany, wesekk@hsu-hh.de

^{**}Faculty of Mathematics and Information Science, Warsaw University of Technology, ul. Koszykowa 75, 00-662 Warszawa, Poland, wesekk@mini.pw.edu.pl

1 Introduction

We deal with the problem of scheduling on two uniform machines: Given are two machines, denoted by M1 and M2, that are both capable to process incoming jobs. They only differ in the processing speed. We assume that machine M1 is working at unit speed 1, and machine M2 is s times faster, with $s \geq 1$. Hence when machine M1 processes a job of length L , then machine M2 can handle this job in L/s time. There is a (finite) number of incoming jobs of various (finite) lengths. The task is to assign the jobs to the two machines. It is desired to finish all incoming jobs as early as possible, that is, to minimize the makespan. In the offline variant of this problem, all jobs to be assigned are fully known in advance. If nothing is known about the jobs beforehand, we are faced with an online problem. We deal here with a semi-online problem, that means, the jobs are still not known individually, but we assume to have some further overall knowledge. In particular, we assume that the value of the solution to the corresponding offline problem, which we denote by OPT, is known in advance.

Let A be an algorithm that solves the semi-online variant of the problem. This algorithm receives, besides the value of OPT, one job after the other in an unknown order and must immediately decide to which of the two machines this job should go to (later changes are not possible). This algorithm A finally arrives at a makespan value M greater than or equal to OPT. Still, M can be compared to OPT by considering the ratio $\frac{M}{\text{OPT}}$. Of course, it is preferred to have an algorithm where this ratio is close to 1 for any given input data. Thus the question arises, how close to 1 can we get? Is it possible to construct an algorithm that reaches this value, or is there a theoretical lower bound well above 1, that no algorithm will ever undercut, no matter how hard it tries?

The paradigm that the instance of a problem is revealed in parts, and the decision has to be made as the part is revealed, is naturally motivated by many real-world applications. As it was mentioned before, if no information is given in advance, then we call such a problem online, and if some partial information about the instance is known beforehand, then the scheme is called semi-online.

The most common way of measuring the quality of an online or semi-online algorithm uses the notion of the mentioned competitive ratio. Assume that we are dealing with a minimization problem and the (offline) optimal value for an instance I is equal to $\text{OPT}(I)$. Formally, an online algorithm is said to be r -competitive if for any instance I the value $A(I)$ of the result of the algorithm satisfies $\frac{A(I)}{\text{OPT}(I)} \leq r$. The competitive ratio of an algorithm is defined as the supremum of such ratios. The question is: what is the best possible ratio for our (online or semi-online) problem? Formally, we would ask for the optimal competitive ratio, that is the infimum over all numbers r for which there exists an r -competitive algorithm. An algorithm is said to be optimal if its competitive ratio matches the best possible lower bound (largest value for which provably no algorithm can have a better competitive ratio).

1.1 Survey of the Literature

The on-line and various semi-online variations of the problem with a set of jobs to be scheduled on m (not necessarily uniform) machines with an objective to minimize the maximum completion time have been studied for decades. Here we will describe some results concerning deterministic algorithms working on *uniformly related* machines, that is, each machine has its speed s and processing a job of length p takes $\frac{p}{s}$ time on it. We assume (except where noted) that any job has to be done on one machine, and jobs arrive in a list one after another (list-online scheme). For information beyond those given below, we refer to the survey of Tan and Zhang [21].

We will first deal with the basic case of identical machines (i.e. with the same speed) in the pure online scheme. A classic work by Graham [15] is probably the first step in this direction. It gives a $(2 - \frac{1}{m})$ -competitive algorithm, by using a heuristic of assigning each task to the currently least loaded machine. It was proved by Faigle et al. [12] that for $m = 2$ and $m = 3$ this algorithm is in fact optimal. In the case of arbitrary m , many papers appeared in order to decrease the gaps between lower and upper bounds. For the lower bound, Gormley et al. [14] showed that no online algorithm can have competitive ratio better than 1.852 generally (when m can be arbitrarily large). For the upper bound, Albers [1] proposed a 1.923-competitive algorithm for any m , moreover Fleischer and Wahl [13] gave an algorithm with competitive ratio tending to a limit near 1.9201 as m gets large.

In the more general case of machines with arbitrary speeds, the best general bounds are the following. Berman et al. [5] provided an 5.828-competitive algorithm, and Ebenlendr and Sgall [9] proved a lower bound of 2.564. If $m = 2$ (and the speeds are 1 and s), then the greedy strategy due to Graham — choosing in every step the machine which will finish the actual job as soon as possible — is again useful. Epstein [11] showed that this algorithm is optimal and has competitive ratio $\min\{1 + \frac{s}{1+s}, 1 + \frac{1}{s}\}$. We will later see that in semi-online variants the situation is much more complex.

How much can we gain if some information is known in advance? In this paper we are interested in the semi-online scheme when only the optimal offline value is known (OPT version, for short), although we want to mention a strong relation with other semi-online version of the described scheduling problem, in which only the sum of jobs is known (SUM version) [3, 2, 8, 17, 18, 20]. Namely, for a given number m of uniform (possibly non-identical) machines the optimal competitive ratio for the OPT version is at most the competitive ratio of the SUM version (see Dósa et al. [8]; for equal speeds it was first implicitly stated by Cheng et al. [6]).

Azar and Regev [4] were the first to investigate the OPT version for identical machines (under the name of bin stretching), although the observation about the relation with the SUM version implies that the first upper bound of $\frac{4}{3}$ for the case of two identical machines follows from the work of Kellerer et al. [17] (now it is known to be optimal). Cheng et al. [6] and Lee and Lim [18] made further progress for the case of more than two identical machines.

Since we are interested in the OPT version with non-identical speeds on two uniform machines, we will state the previous results in terms of s (recall that the speeds are 1 and s). Epstein [11] was first to investigate this problem. She proved the following bounds for the optimal competitive ratio $r^*(s)$:

$$r^*(s) : \begin{cases} r^*(s) \in \left[\frac{3s+1}{3s}, \frac{2s+2}{2s+1} \right] & \text{for } s \in [1, q_E \approx 1.1243] \\ r^*(s) \in \left[s \left(\frac{3}{4} + \frac{\sqrt{65}}{20} \right), \frac{2s+2}{2s+1} \right] & \text{for } s \in [q_E, \frac{1+\sqrt{65}}{8} \approx 1.1328] \\ r^*(s) = \frac{2s+2}{2s+1} & \text{for } s \in \left[\frac{1+\sqrt{65}}{8}, \frac{1+\sqrt{17}}{4} \approx 1.2808 \right] \\ r^*(s) = s & \text{for } s \in \left[\frac{1+\sqrt{17}}{4}, \frac{1+\sqrt{3}}{2} \approx 1.3660 \right] \\ r^*(s) \in \left[\frac{2s+1}{2s}, s \right] & \text{for } s \in \left[\frac{1+\sqrt{3}}{2}, \sqrt{2} \approx 1.4142 \right] \\ r^*(s) \in \left[\frac{2s+1}{2s}, \frac{s+2}{s+1} \right] & \text{for } s \in \left[\sqrt{2}, \frac{1+\sqrt{5}}{2} \approx 1.6180 \right] \\ r^*(s) \in \left[\frac{s+1}{2}, \frac{s+2}{s+1} \right] & \text{for } s \in \left[\frac{1+\sqrt{5}}{2}, \sqrt{3} \approx 1.7321 \right] \\ r^*(s) = \frac{s+2}{s+1} & \text{for } s \geq \sqrt{3} \end{cases}$$

where q_E is the solution of $36x^4 - 135x^3 + 45x^2 + 60x + 10 = 0$. It means that her bounds are proven optimal on some intervals, where the upper and lower bounds coincide. It will turn out, however, that some of her bounds are also optimal, where she finds a lower bound (but her upper bound needs to be decreased), or she finds an upper bound (but her lower bound needs to be increased). On two intervals the question about the tight bound remained open after the work of Epstein. These are $[1, \frac{1+\sqrt{17}}{4} \approx 1.2808]$ and $[\frac{1+\sqrt{3}}{2} \approx 1.3660, \sqrt{3}]$, which will be called *left interval* and *right interval*, respectively.

Ng et al. [20] considered the *right interval*. They presented algorithms giving the upper bounds

$$r^*(s) \leq \begin{cases} \frac{2s+1}{2s} & \text{for } s \in \left[\frac{1+\sqrt{3}}{2}, \frac{1+\sqrt{21}}{4} \approx 1.3956 \right] \\ \frac{6s+6}{4s+5} & \text{for } s \in \left[\frac{1+\sqrt{21}}{4}, \frac{1+\sqrt{13}}{3} \approx 1.5352 \right] \\ \frac{12s+10}{9s+7} & \text{for } s \in \left[\frac{1+\sqrt{13}}{3}, \frac{5+\sqrt{241}}{12} \approx 1.7104 \right] \\ \frac{2s+3}{s+3} & \text{for } s \in \left[\frac{5+\sqrt{241}}{12}, \sqrt{3} \right] \end{cases}$$

and provided the following lower bounds:

$$r^*(s) \geq \begin{cases} \frac{3s+5}{2s+4} & \text{for } s \in \left[\sqrt{2}, \frac{\sqrt{21}}{3} \approx 1.5275 \right] \\ \frac{3s+3}{3s+1} & \text{for } s \in \left[\frac{\sqrt{21}}{3}, \frac{5+\sqrt{193}}{12} \approx 1.5744 \right] \\ \frac{4s+2}{2s+3} & \text{for } s \in \left[\frac{5+\sqrt{193}}{12}, \frac{7+\sqrt{145}}{12} \approx 1.5868 \right] \\ \frac{5s+2}{4s+1} & \text{for } s \in \left[\frac{7+\sqrt{145}}{12}, \frac{9+\sqrt{193}}{14} \approx 1.6352 \right] \\ \frac{7s+4}{7s} & \text{for } s \in \left[\frac{9+\sqrt{193}}{14}, \frac{5}{3} \right] \\ \frac{7s+4}{4s+5} & \text{for } s \in \left[\frac{5}{3}, \frac{5+\sqrt{73}}{8} \approx 1.6930 \right] \end{cases}$$

Hence from the work of Ng et al. it turned out that Epstein's lower bound is in fact tight on the very left end of the right interval.

Finally, Dósa et al. [8] considered the *left interval* and solved the problem there almost completely, by providing the following bounds:

$$r^*(s) \geq \begin{cases} \frac{8s+5}{5s+5} & \text{for } s \in \left[\frac{5+\sqrt{205}}{18}, \frac{1+\sqrt{31}}{6} \approx 1.0946 \right] \\ \frac{2s+2}{2s+1} & \text{for } s \in \left[\frac{1+\sqrt{31}}{6}, \frac{1+\sqrt{17}}{4} \approx 1.2808 \right] \end{cases}$$

$$r^*(s) \leq \begin{cases} \frac{3s+1}{3s} & \text{for } s \in [1, q_D \approx 1.071] \\ \frac{7s+6}{4s+6} & \text{for } s \in [q_D, \frac{1+\sqrt{145}}{12} \approx 1.0868] \end{cases}$$

where q_D is the unique root of equation $3s^2(9s^2 - s - 5) = (3s + 1)(5s + 5 - 6s^2)$.

Altogether it means that Epstein's lower bound is tight on the left end of the left interval, and her upper bound is tight on the right end of the left interval, but the problem about the optimal value remained open in the middle for a small interval. For a visual summary (with our contribution included), see Figures 1–4. Whenever the dotted line (that represents an upper bound) is on an unbroken line (that represents a lower bound), the optimal competitive ratio is known.

As the problem is almost completely solved on the left interval, we revisit the right interval. As a consequence of earlier results, before this publication the optimality question was unsettled on the interval $(\frac{1+\sqrt{21}}{4}, \sqrt{3}) \approx (1.3956, 1.7321)$.

There are also two variations of the aforementioned problems we find worth mentioning, as they have been studied also in the case of OPT version. In the scheduling with *preemption* model it is allowed to split a job into multiple parts and assign them to different machines, as long as those parts will be processed in disjoint time intervals. Ebenlendr and Sgall [10] presented one optimal preemptive algorithm working for various semi-online conditions and their combinations, including OPT version, SUM version, known longest job, jobs sorted in decreasing order; this many-sided optimality seems to be much different from the situation in models without preemption. In the second variation, called *under a grade of service provision* (GoS), not every job can be processed by every machine: there is a level function for the jobs, and a job with level i can be processed only by machines with index at most i . The GoS-OPT version for two (nonidentical) machines is also solved: Lu and Liu [19] proved that the optimal competitive ratio is $\min\{\frac{1+2s}{1+s}, \frac{1+s}{s}\}$. This is again equal to the optimal competitive ratio of the GoS-SUM version.

1.2 Our Contribution

Expressing in general terms, we present a rather complicated construction of malicious input sequences for a scheduling problem, which provides the best possible lower bound on the competitive ratio of a class of semi-online algorithms. The novelty of our method compared to earlier approaches in the literature is that it goes several levels deeper. This idea may turn out to be useful towards the solution of further open problems in the area.

More explicitly, we deal with the semi-online two uniform machines scheduling problem with a “known opt” condition, that is, $Q_2|\text{OPT}|C_{max}$ according to three-field notation introduced by Graham et al. [16]. This problem is studied on two parts of the right interval, $[\frac{1+\sqrt{21}}{4}, \frac{3+\sqrt{73}}{8}] \approx [1.3956, 1.443]$ and $[\frac{5}{3}, \frac{4+\sqrt{133}}{9}] \approx [\frac{5}{3}, 1.7258]$, for which we give new lower-bound constructions. (Note that these numbers are solutions of certain equations, and will be formally introduced in the following section.) We apply an adversary strategy, that is, depending on the current assignment of a given job the adversary defines the next job that makes life complicated for the algorithm. We show that the input can always be continued in such a malicious way that any kind of algorithm will exceed the lower bound on the makespan at some step.

“Marry, and you will regret it; don’t marry, you will also regret it; marry or don’t marry, you will regret it either way” — says the Danish philosopher Søren Kierkegaard. He describes a decision situation from which two different possible choices will lead into the future. But no matter how the person decides, the continuation will not lead to a happy end. Both can be seen as “unhappy situations” (from the person’s point of view who has to make the decision). This in our scheduling setting will correspond to the three “Final Cases” that are described in Section 2.3. The scheduler (the algorithm) can make a decision, similar to the person deciding for or against a marriage. No matter how the algorithm decides, it will lead to an “unhappy situation”, since it is possible to generate further jobs, based on its decision, in such a way that the competitive ratio of the algorithm is relatively high. It will not be obvious at first sight that the algorithm is always trapped in a situation that leads to an unhappy ending; it evolves over several rounds of further jobs that are determined in what we call “intermediate cases” or “final cases”. At most, eight jobs need to be generated to close the trap.

The moves of the algorithm and the generation of jobs can be seen as a two-player game, such as chess. One player is the algorithm, and the other player is the adversary that constructs malicious jobs. What our result then shows is that the player of the algorithm is “checkmate” after at most eight moves. Our purpose was not to give a difficult construction, however, we were not able to get the tight bounds with less than eight jobs.

Together with the upper-bound result of Ng et al. [20] we obtain that our new lower bound is tight (that is, the algorithms presented by Ng et al. are optimal) for any value of s in one of these intervals: $[\frac{1+\sqrt{21}}{4}, \frac{3+\sqrt{73}}{8}] \approx [1.3956, 1.443]$, $[\frac{5}{3}, \frac{13+\sqrt{1429}}{30}] \approx [\frac{5}{3}, 1.6934]$, and $[\frac{31+\sqrt{8305}}{72}, \frac{5+\sqrt{241}}{12}] \approx [1.6963, 1.7103]$, see also Figure 1.

Between $\frac{5}{3}$ and $\sqrt{3}$ there remain two intervals where the question for an optimal algorithm remains open: the first interval between approximately 1.6934 and 1.6963, see Figure 4, that we call *narrow interval*, and the second interval between approximately 1.7103 and $\sqrt{3}$ that we call *wide interval*. In case of the narrow interval, the new lower bound is very close to the existing upper bound, so here we do not try to eliminate the small gap (which is smaller than 0.000209). But in case of the wide interval the situation is different: In our

subsequent paper [7] we prove that our new lower bound is optimal in the left part of the wide interval. Furthermore, it turns out that Epstein's previous lower bound is optimal in the right side of the wide interval, see Figure 3. Therefore, together with the result proven in [7], with the exception of the narrow interval we completely solve the case of any speed in $[\frac{5}{3}, \sqrt{3}]$.

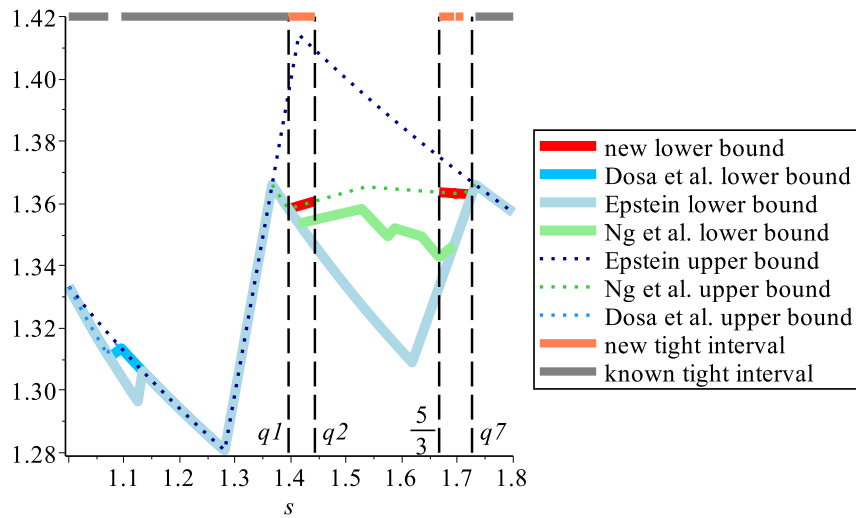


Figure 1: Our new lower bound in comparison with existing lower and upper bounds from Epstein [11], Ng et al. [20], and Dósa et al. [8].

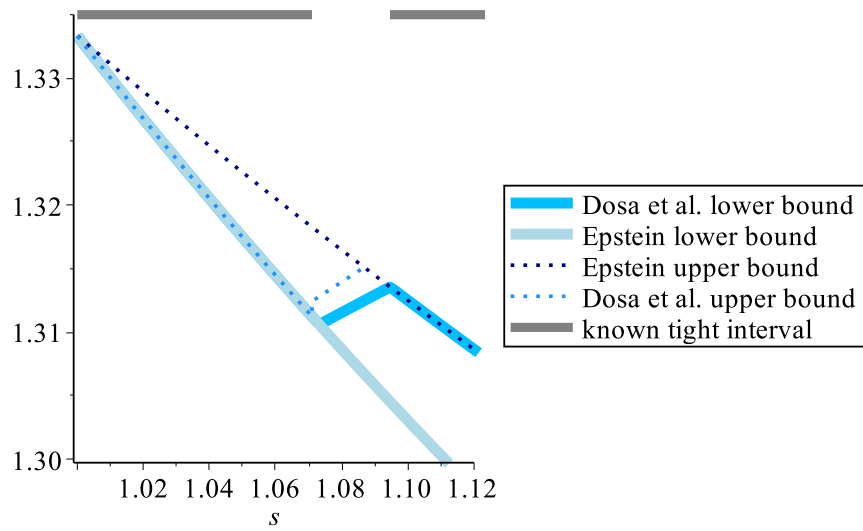


Figure 2: Zooming into the left part of Figure 1.

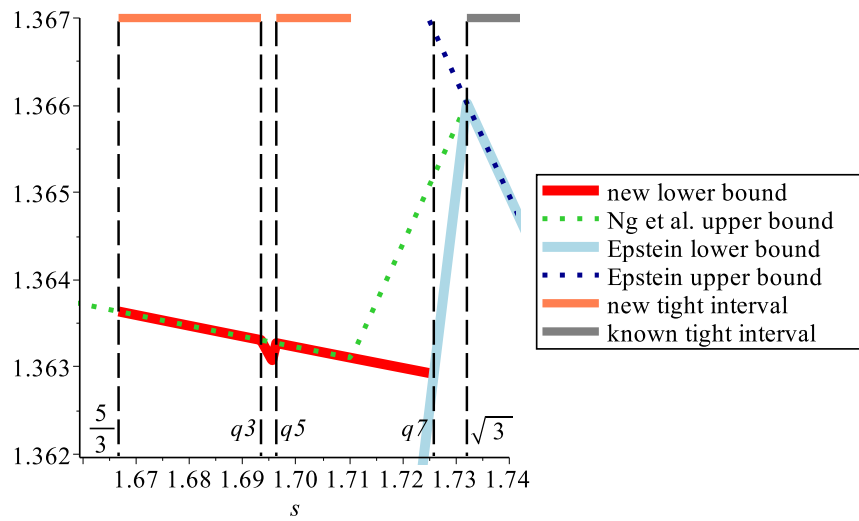


Figure 3: Zooming into the right part of Figure 1.

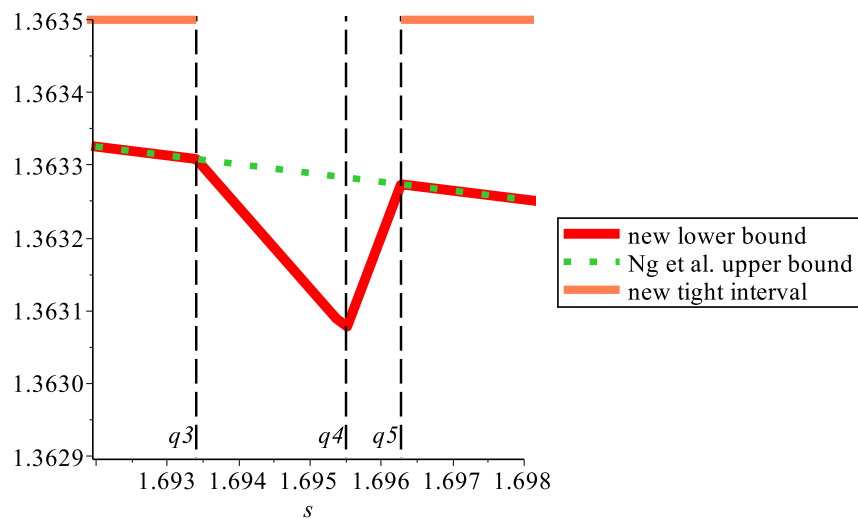


Figure 4: Zooming into the middle part of Figure 3.

2 Preliminaries and Notations

Let OPT and SUM mean, respectively, the known optimum value (given by some oracle), and the total size of the jobs. By \mathcal{J}_t we denote the family of all sets of jobs with optimum value equal to t . In other words, if the oracle returns a value of OPT , then there is a guarantee that the set of jobs belongs to \mathcal{J}_{OPT} . We denote the prescribed competitive ratio (that we do not want to violate) by r . The *load* of a machine means the sum of processing times of jobs being assigned there during the scheduling procedure so far. The loads of the machines will sometimes be mentioned in the text as M1-load and M2-load, respectively.

Lemma 1 *Given a set of jobs, assume that it is possible to assign them to the two machines in such way that M1 receives a load of t and M2 a load of $s \cdot t$. Then this set of jobs belongs to \mathcal{J}_t .*

Proof. The assignment of jobs given in the formulation of the lemma is a feasible solution with a makespan of t . It remains to show that there is no better solution. The sum of jobs (the total load) is $(s+1) \cdot t$. Assume that there is a better assignment with makespan $t' < t$. Then the load on machine M1 can be at most t' . Hence the load on machine M2 is the remaining load, which is at least $st + (t - t') > st > st'$. This contradicts the assumption that t' is the makespan. ■

As a consequence of Lemma 1, we remark that $\text{SUM} \leq (s+1) \cdot \text{OPT}$, and that the size of any job is at most $s \cdot \text{OPT}$. We denote $\overline{\text{SUM}} := (s+1) \cdot \text{OPT}$.

2.1 Definitions

Let $q_1 := \frac{\sqrt{21}+1}{4} \approx 1.3956$, which is the positive solution of $\frac{2s+1}{2s} = \frac{6s+6}{4s+5}$.

Let $q_2 := \frac{\sqrt{73}+3}{8} \approx 1.443$, which is the positive solution of $\frac{6s+6}{4s+5} = \frac{5s+2}{4s+1}$.

Let $q_3 := \frac{13+\sqrt{1429}}{30} \approx 1.6934$, which is the positive solution of $\frac{12s+10}{9s+7} = \frac{18s+16}{16s+7}$.

Let $q_4 := \frac{30+7\sqrt{186}}{74} \approx 1.6955$, which is the positive solution of $\frac{18s+16}{16s+7} = \frac{8s+7}{3s+10}$.

Let $q_5 := \frac{31+\sqrt{8305}}{72} \approx 1.6963$, which is the positive solution of $\frac{8s+7}{3s+10} = \frac{12s+10}{9s+7}$.

Let $q_6 := \frac{5+\sqrt{241}}{12} \approx 1.7103$, which is the positive solution of $\frac{12s+10}{9s+7} = \frac{2s+3}{s+3}$.

Let $q_7 := \frac{4+\sqrt{133}}{9} \approx 1.7258$, which is the positive solution of $\frac{12s+10}{9s+7} = \frac{s+1}{2}$.

We note that in this paper we do not consider speeds between q_2 and $\frac{5}{3}$. Now we define the ratio for our new lower bound. It differs on several intervals, where the speed is called *small*, *regular*, or *medium*. The latter two are further divided as smaller or bigger medium, and smaller or bigger regular, as follows.

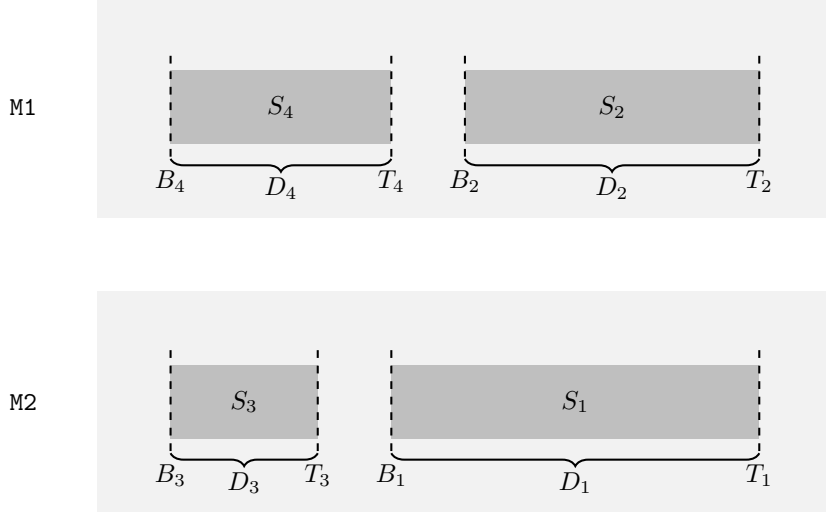


Figure 5: Safe sets. The horizontal axis is reflecting the time.

Let

$$r(s) = \begin{cases} r_1(s) := \frac{6s+6}{4s+5} & \text{if } q_1 \leq s \leq q_2 \approx 1.443 & \text{i.e. } s \text{ is small} \\ r_2(s) := \frac{12s+10}{9s+7} & \text{if } \frac{5}{3} \leq s \leq q_3 \approx 1.6934 & \text{i.e. } s \text{ is smaller regular} \\ r_3(s) := \frac{18s+16}{16s+7}, & \text{if } q_3 \leq s \leq q_4 \approx 1.6955 & \text{i.e. } s \text{ is smaller medium} \\ r_4(s) := \frac{8s+7}{3s+10}, & \text{if } q_4 \leq s \leq q_5 \approx 1.6963 & \text{i.e. } s \text{ is bigger medium} \\ r_2(s) := \frac{12s+10}{9s+7} & \text{if } q_5 \leq s \leq q_7 \approx 1.7258 & \text{i.e. } s \text{ is bigger regular} \end{cases}$$

As we will show in the very end (cf. Theorem 16), this function will be our lower bound on the optimal competitive ratio.

As abbreviations, we call s *regular* if s is smaller regular or bigger regular, and we call s *medium* if s is smaller medium or bigger medium.

Now we define the so called “safe sets”. A safe set is a time interval on some of the machines, and it is safe in the sense that if the load of the machine is in this interval, this enables a “smart” algorithm to finish the schedule by not violating the desired competitive ratio. In other words, from the point of view of a lower-bound construction (the adversary), we should avoid that an algorithm can assign the actual job in a way that the increased load of some machine will be inside a safe set.

Safe sets S_1 and S_3 are defined on the second machine, while safe sets S_2 and S_4 are defined on the first machine. We introduce notations for the top, bottom, and length of the safe sets. Thus let $S_i = [B_i, T_i]$, and $D_i = T_i - B_i$, for any $1 \leq i \leq 4$, see Figure 5. We will show in Section 2.2 that these intervals are well-defined.

Recall that the optimum value is known. For the sake of simplicity let

us assume without loss of generality that $\text{OPT} = 1$. (This can be done by normalization: If OPT differs from unit, one can divide all job sizes by the value of OPT .)

Then the boundaries of safe sets S_i are defined as follows.

1. $B_1 = s + 1 - r$, and $T_1 = rs$, thus $D_1 = (s + 1)(r - 1)$;
2. $B_2 = s + 1 - sr$, and $T_2 = r$, thus $D_2 = (s + 1)(r - 1)$;
3. $B_3 = 2s - 2r - rs + 2$, and $T_3 = s(r - 1)$, thus $D_3 = 2r - 3s + 2rs - 2$;
4. $B_4 = 4s - 2r - 3rs + 3$, and $T_4 = r - 1$, thus $D_4 = (3r - 4)(s + 1)$.

We introduce as abbreviations some expressions that are used in the sequel.

$$\begin{aligned} a &:= T_4 - B_3, \\ b &:= T_3 - B_2, \\ c &:= 1 - D_1, \end{aligned}$$

and, if $s \geq \frac{5}{3}$, let

$$d := b - c - B_4.$$

For bigger regular speeds, we will also need the next notations:

$$\begin{aligned} e &:= \frac{1}{2}(b - 2c - a - B_4), \\ f &:= \frac{1}{2}(a + b - B_4), \\ g &:= \frac{1}{2}(b - a - B_4). \end{aligned}$$

By easy calculations we get the following expressions:

$$\begin{aligned} a &= rs + 3r - 2s - 3, \\ b &= 2rs - 2s - 1, \\ c &= -rs - r + s + 2, \\ d &= 6rs + 3r - 7s - 6. \end{aligned}$$

Note that

$$\begin{aligned} a + b &= D_4, \\ e + f &= d, \\ f + g &= b - B_4, \\ f - g &= a, \\ c + e &= g. \end{aligned}$$

Moreover, for the values of e , f , and g , we get the following equalities:

$$\begin{aligned} e &= \frac{1}{2}r - 3s + 3rs - \frac{5}{2}, \\ f &= \frac{5}{2}r - 4s + 3rs - \frac{7}{2}, \\ g &= 2rs - 2s - \frac{1}{2}r - \frac{1}{2}. \end{aligned}$$

2.2 General Properties

In Figure 6 we show plots of the functions r_2 , r_3 , and r_4 . If s is medium, then both $r_3(s)$ and $r_4(s)$ are below $r_2(s)$. Moreover $r_3(s) \geq r_4(s)$ if s is small medium, and the opposite inequality holds if s is bigger medium. Note that s is medium sized only on the narrow interval.

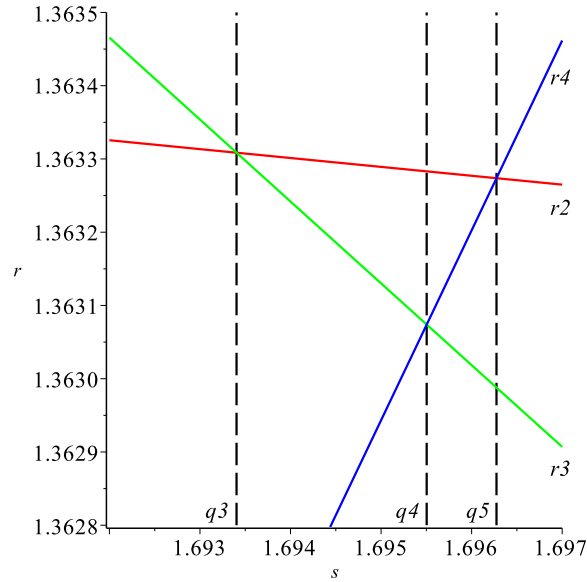


Figure 6: Comparing $r_2(s)$ (red), $r_3(s)$ (green), and $r_4(s)$ (blue).

- Lemma 2**
1. $r_1(s) \leq r_2(s)$ if $s \leq q_2$,
 2. $r_2(s) \leq r_1(s)$ if $s \geq \frac{5}{3}$,
 3. $r_3(s) \leq r_2(s)$ if $s \geq q_3$,
 4. $r_4(s) \leq r_2(s)$ if $s \leq q_5$.

Proof.

1. This estimation was already proven in [20]. We repeat it here for the sake of completeness. The inequality $r_1(s) \leq r_2(s)$ is equivalent to $(12s + 10)(4s + 5) - (6s + 6)(9s + 7) = -6s^2 + 4s + 8 \geq 0$, which holds if and only if $\frac{1-\sqrt{13}}{3} \leq s \leq \frac{1+\sqrt{13}}{3} \approx 1.5352$. Hence it holds for all $s \leq q_2$.
2. This follows from the previous computations.
3. The inequality $r_3(s) \leq r_2(s)$ is equivalent to $(12s + 10)(16s + 7) - (18s + 16)(9s + 7) = 30s^2 - 26s - 42 \geq 0$, which holds if and only if $s \leq \frac{13-\sqrt{1429}}{30}$ or $s \geq \frac{13+\sqrt{1429}}{30} = q_3$.
4. The inequality $r_4(s) \leq r_2(s)$ is equivalent to $(12s + 10)(3s + 10) - (8s + 7)(9s + 7) = -36s^2 + 31s + 51 \geq 0$, which holds if and only if $\frac{31-\sqrt{8305}}{72} \leq s \leq \frac{31+\sqrt{8305}}{72} = q_5$.

■

In the next lemma we prove lower and upper bounds on $r(s)$. These bounds are needed to show that the safe sets are well-defined.

- Lemma 3**
1. $\frac{3s+2}{2s+2} < \frac{4}{3} < 1.35 < r(s) < \min\left\{\frac{4s+3}{3s+2}, \frac{s+2}{s+1}\right\} < \frac{2s+2}{s+2} < \frac{2s+1}{s+1}$
hold in the entire considered domain of the function r , i.e., for all $s \in [q_1, q_2] \cup [\frac{5}{3}, q_7] =: \text{Dom}(r)$.
 2. *If $s \geq \frac{5}{3}$, we have $r(s) \geq \frac{8s+7}{6s+5}$.*

Proof.

1. The leftmost lower bound holds as $\frac{3s+2}{2s+2} < \frac{4}{3}$ is equivalent to $4(2s + 2) - 3(3s + 2) > 0$, i.e. $s < 2$.

Now we show that $r(s) > 1.35$.

- For $q_1 \leq s \leq q_2$ where $r(s) = r_1(s)$, we get $0 < \frac{6s+6}{4s+5} - \frac{135}{100} = \frac{12s-15}{80s+100}$, which is true since $s > \frac{5}{4}$.
- For $\frac{5}{3} \leq s \leq q_3$ or $q_5 \leq s \leq q_6$, where $r(s) = r_2(s)$, we similarly obtain $0 < \frac{12s+10}{9s+7} - \frac{135}{100} = \frac{11-3s}{20(9s+7)}$, which is true since $s < \frac{11}{3}$.
- For $q_3 \leq s \leq q_4$ where $r(s) = r_3(s)$, we get $0 < \frac{18s+16}{16s+7} - \frac{135}{100} = \frac{131-72s}{20(16s+7)}$, which is true since $s < \frac{131}{72} \approx 1.8194$.
- For $q_4 \leq s \leq q_5$ where $r(s) = r_4(s)$, we get $0 < \frac{8s+7}{3s+10} - \frac{135}{100} = \frac{79s-130}{20(3s+10)}$, which is true since $s > \frac{130}{79} \approx 1.6456$.

Hence $r(s) > 1.35$.

Regarding the rightmost upper bound, $\frac{2s+2}{s+2} < \frac{2s+1}{s+1}$ holds since $\frac{2s+1}{s+1} - \frac{2s+2}{s+2} = \frac{s}{(s+2)(s+1)} > 0$.

Moreover, $\frac{2s+2}{s+2} - \frac{4s+3}{3s+2} = \frac{2s^2-s-2}{(s+2)(3s+2)} > 0$, which holds since $2s^2 - s - 2 > 0$ for all $s < \frac{1-\sqrt{17}}{4}$ or $s > \frac{1+\sqrt{17}}{4} \approx 1.2808$. Thus, it holds that $\min \left\{ \frac{4s+3}{3s+2}, \frac{s+2}{s+1} \right\} < \frac{2s+2}{s+2}$.

It remains to show that $r < \min \left\{ \frac{s+2}{s+1}, \frac{4s+3}{3s+2} \right\}$. Note that $\frac{4s+3}{3s+2} \leq \frac{s+2}{s+1}$ holds for all $s \in \text{Dom}(r)$, if and only if $(4s+3)(s+1) - (s+2)(3s+2) = s^2 - s - 1 \leq 0$, i.e., $\frac{1-\sqrt{5}}{2} \leq s \leq \frac{1+\sqrt{5}}{2} \approx 1.618$. Thus, if $s \leq q_2$, then we need to verify that $r < \frac{4s+3}{3s+2}$; otherwise, if $s \geq \frac{5}{3}$, we need to verify that $r < \frac{s+2}{s+1}$.

For $q_1 \leq s \leq q_2$ where $r(s) = r_1(s)$, we get $\frac{4s+3}{3s+2} - \frac{6s+6}{4s+5} = \frac{2s+3-2s^2}{(4s+5)(3s+2)} > 0$, which holds since $2s+3-2s^2 > 0$ for all $\frac{1-\sqrt{7}}{2} < s < \frac{1+\sqrt{7}}{2} \approx 1.8229$.

Now let us consider the range $\frac{5}{3} \leq s \leq q_6$. For the two cases where $r(s) = r_2(s)$ we have that $\frac{12s+10}{9s+7} < \frac{s+2}{s+1}$ holds since $\frac{s+2}{s+1} - \frac{12s+10}{9s+7} = \frac{-3s^2+3s+4}{(s+1)(9s+7)} > 0$, which is valid since $-3s^2+3s+4 > 0$ for all $\frac{1}{2} - \frac{\sqrt{57}}{6} < s < \frac{1}{2} + \frac{\sqrt{57}}{6} \approx 1.7583$.

In Lemma 2 we showed that $r_3(s) \leq r_2(s)$ for all $s \geq q_3$ and that $r_4(s) \leq r_2(s)$ for all $s \leq q_5$. From this the claimed upper bound on $r(s)$ follows.

2. For $r(s) = r_2(s)$ we get $\frac{12s+10}{9s+7} - \frac{8s+7}{6s+5} = \frac{s+1}{(6s+5)(9s+7)} > 0$.

For $r(s) = r_3(s)$ we get $\frac{18s+16}{16s+7} - \frac{8s+7}{6s+5} = \frac{31+18s-20s^2}{(6s+5)(16s+7)} > 0$ for all $s < \frac{9+\sqrt{701}}{20} \approx 1.7738$.

For $r(s) = r_4(s)$ we get $\frac{8s+7}{3s+10} - \frac{8s+7}{6s+5} = \frac{(8s+7)(3s-5)}{(3s+10)(6s+5)} \geq 0$, because $s \geq \frac{5}{3}$.

■

Now we derive several properties regarding the boundaries of the safe sets. Note that the safe sets could have been defined in an alternative way as sets satisfying the next properties.

Lemma 4 1. $D_1 = D_2$,

2. $T_1 - T_3 = s$ and $T_2 - T_4 = 1$,

3. $B_3 = B_1 - D_1$,

4. $B_4 = B_2 - D_3$,

5. $B_2 + T_3 = 1$,

6. $T_1 + B_2 = T_2 + B_1 = \overline{SUM}$.

Proof. All properties are checked using the definition of the safe sets.

1. $D_1 = D_2$ holds by definition.

2. $T_1 - T_3 = rs - s(r-1) = s$ and $T_2 - T_4 = r - (r-1) = 1$.
3. $B_3 + D_1 = (2s - 2r - rs + 2) + (s+1)(r-1) = s - r + 1 = B_1$.
4. $B_4 + D_3 = (4s - 2r - 3rs + 3) + (2r - 3s + 2rs - 2) = s + 1 - rs = B_2$.
5. $B_2 + T_3 = (s + 1 - sr) + s(r-1) = 1$.
6. $T_1 + B_2 = rs + (s + 1 - sr) = s + 1 = \overline{\text{SUM}}$. Moreover $T_2 + B_1 = r + (s + 1 - r) = s + 1 = \overline{\text{SUM}}$.

■

Now we show that the definition of the safe sets is of sense, these sets do not intersect each other, and they follow each other on the machines.

Lemma 5 1. $0 < B_3 < T_3 < B_1 < T_1$.

2. $0 < B_4 < T_4 < B_2 < T_2$.

Proof. In the calculations we generally use Lemma 3, unless stated otherwise.

1. From $r < \frac{2s+2}{s+2}$ it follows that $0 < 2s - 2r - rs + 2 = B_3$. From $r > \frac{3s+2}{2s+2}$ it follows that $0 < 2r + 2rs - 3s - 2 = D_3 = T_3 - B_3$. From $r < \frac{2s+1}{s+1}$ it follows that $0 < (s+1-r) - s(r-1) = B_1 - T_3$. From the definition we have that $0 < (s+1)(r-1) = D_1 = T_1 - B_1$.
2. From $r < \frac{4s+3}{3s+2}$ it follows that $0 < 4s + 3 - 3rs - 2r = B_4$. From $r > \frac{4}{3}$ and the definition we have that $0 < (3r-4)(s+1) = D_4 = T_4 - B_4$. From $r < \frac{s+2}{s+1}$ it follows that $0 < (s+1-sr) - (r-1) = B_2 - T_4$. From the definition we have that $0 < (s+1)(r-1) = D_2 = T_2 - B_2$.

■

Now we have seen that the safe sets are properly defined. We will need some further bounds on r . With their help, we shall be able to prove several properties of the expressions we introduced.

Lemma 6 *The following bounds on r are valid:*

1. $\frac{2s+1}{2s} \leq r < s$.
2. If $s \geq \frac{5}{3}$, then $\frac{7s+6}{6s+3} \leq r \leq \frac{3s+5}{2s+4}$.
3. If s is regular, then we have $\frac{6s+5}{6s+1} \leq r$.

Proof.

1. This bound was already proven in [20] (cf. Figure 1 in [20]). We give it here for the sake of completeness. Regarding the lower bound, if s is small (i.e., $q_1 \leq s \leq q_2$), then $\frac{2s+1}{2s} \leq r = r_1(s) = \frac{6s+6}{4s+5}$ holds, since $2s(6s+6) - (2s+1)(4s+5) = 4s^2 - 2s - 5 \geq 0$, if $s \geq \frac{1+\sqrt{21}}{4} = q_1$. For $s \geq \frac{5}{3}$, we know from Lemma 3 that $\frac{4}{3} < r$, and thus get $\frac{4}{3} - \frac{2s+1}{2s} = \frac{2s-3}{6s} > 0$. Now let us consider the upper bound. For a small speed ratio s , we get $s - \frac{6s+6}{4s+5} = \frac{4s^2-s-6}{4s+5} > 0$ if $s > \frac{1+\sqrt{97}}{8} \approx 1.3561$, in particular for $s \geq q_1$. For $s \geq \frac{5}{3}$, the statement follows from Lemma 2, since $r_1(s) \geq r_2(s)$ for these speeds.
2. Let us consider the lower bound. In the case $r = r_2(s)$, we get $\frac{12s+10}{9s+7} - \frac{7s+6}{6s+3} = \frac{9s^2-7s-12}{3(9s+7)(2s+1)} \geq 0$ if $s \geq \frac{7+\sqrt{481}}{18} \approx 1.6073$. In the case $r = r_3(s)$, we get $\frac{18s+16}{16s+7} - \frac{7s+6}{6s+3} = \frac{(4s+3)(2-s)}{3(2s+1)(16s+7)} > 0$. Finally, in the case $r = r_4(s)$, we get $\frac{8s+7}{3s+10} - \frac{7s+6}{6s+3} = \frac{27s^2-22s-39}{3(3s+10)(2s+1)} \geq 0$ if $s \geq \frac{11+\sqrt{1174}}{27} \approx 1.6764$, which in particular holds for $s \geq q_4$. To prove the upper bound, it suffices to show that $r_2(s) \leq \frac{3s+5}{2s+4}$, since $r \leq r_2(s)$ by Lemma 2. We get $\frac{3s+5}{2s+4} - \frac{12s+10}{9s+7} = \frac{(s+1)(3s-5)}{2(s+2)(9s+7)} \geq 0$.
3. If s is regular, we get $\frac{12s+10}{9s+7} - \frac{6s+5}{6s+1} = \frac{(6s+5)(3s-5)}{(9s+7)(6s+1)} \geq 0$.

■

In the next lemma we prove several properties. Note that d is defined only if $s \geq \frac{5}{3}$, and e, f, g are defined only if s is bigger regular.

Lemma 7 1. $b, c, d, e, f, g \geq 0$,

2. If $s \geq \frac{5}{3}$, then $a \leq c$,

3. $s > T_2$,

4. $B_1 > 1$.

Proof.

1. In the proof we use Lemma 3 and Lemma 6.

- $b \geq 0$, since $r \geq \frac{2s+1}{2s}$.
- $c > 0$, since $r < \frac{s+2}{s+1}$.
- $d \geq 0$, since $r \geq \frac{7s+6}{6s+3}$.
- $e > 0$, since $r > \frac{6s+5}{6s+1}$. The strict inequality follows from the proof of Lemma 6, because $e = 0$ can only occur for $s = \frac{5}{3}$, which means that s would be smaller regular.
- $f > 0$, since $r > \frac{8s+7}{6s+5}$. The strict inequality can also be seen in the proof of Lemma 3, since f is defined only for bigger regular speeds.

- $g > 0$, since $g = c + e > 0$.
- 2. If $s \geq \frac{5}{3}$, then $a \leq c$, since $c - a = (s - r - rs + 2) - (3r - 2s + rs - 3) = 3s - 4r - 2rs + 5 \geq 0$, where the last estimation uses $r \leq \frac{3s+5}{2s+4}$.
- 3. We obtain $s - T_2 = s - r > 0$, since $r < s$ from Lemma 6.1.
- 4. $B_1 - 1 = (s + 1 - r) - 1 = s - r > 0$, since $r < s$ by Lemma 6.1.

■

In fact, we did not “forget” to prove $a \geq 0$, as the following remark shows.

Remark 8 *If $q_6 < s \leq q_7$, then a is negative.*

Proof. Here s is bigger regular, hence $r = r_2(s) = \frac{12s+10}{9s+7}$. Substituting this expression, we thus get $a = 3r - 2s + rs - 3 = \frac{-6s^2+5s+9}{9s+7} < 0$. ■

2.3 General Subcases

The general idea to show that no semi-online algorithm knowing s and OPT can in general be better than the ratio $r(s)$ is to construct a malicious sequence of jobs that is in \mathcal{J}_{OPT} but forces any algorithm to schedule them in such way that they have a makespan of at least r . We construct this sequence iteratively, depending on the previous assignment choices of the algorithm. This leads to a number of cases that need to be considered separately. Some cases are “final” in the next sense (as endgame in chess): When entering the case, the algorithm is trapped, because we can then construct one or a few more jobs, which make the algorithm to overshoot the desired makespan.

In this sense, the following three “G”-cases, G1, G2, and G3, are general, because most of the other cases, independent of the particular interval of s , will lead to them.

We will denote by L_1 and L_2 the current load of machine M1 and M2, respectively. Moreover let L'_1 and L'_2 denote the increased load of that machine, if the actual job is assigned there.

Final Case G1. Suppose $T_3 \leq L_2$ and $L_1 + L_2 \leq 1$.

Note that $1 = B_2 + T_3 > T_3$ holds by Lemma 4.5. In this situation, let the next job be $A = s$ and $B = 1 - (L_1 + L_2) \geq 0$. If A is assigned to M2, we get $L'_2 \geq T_3 + s = T_1$ by Lemma 4.2, thus we are done. Otherwise A is assigned to M1. Then $L'_1 \geq s > T_2$, by Lemma 7.3, and we are done again.

The set of all jobs that were initially on M1 and M2, plus jobs A and B belong to \mathcal{J}_{OPT} by Lemma 1, if we assign A to M2 and all other jobs to M1.

Final Case G2. Suppose $L_1 = B_2$ and $0 \leq L_2 \leq B_3$, and there exists an already assigned job with size c (where c as defined above).

The next and last jobs are $B = D_1$ and

$$\begin{aligned} C &= \overline{\text{SUM}} - (L_1 + L_2 + B) \geq (T_2 + B_1) - (B_2 + D_2) - B_3 \\ &= B_1 - B_3 = D_1 = B \end{aligned}$$

using $\overline{\text{SUM}} = T_2 + B_1$, $D_1 = D_2$, and $B_1 - B_3 = D_1$ from Lemmas 4.1, 4.3, and 4.6. If any of B and C is assigned to M1, the lower bound holds since $L'_1 \geq B_2 + D_1 = B_2 + D_2 = T_2$. Otherwise both jobs go to M2 and we are done again as $L'_2 = \overline{\text{SUM}} - B_2 = T_1$ by Lemma 4.6.

The set of jobs belong to \mathcal{J}_{OPT} by Lemma 1. Indeed, assign jobs c and $B = D_1$ to machine M1, then $L'_1 = c + D_1 = 1$ by the definition of c . All remaining jobs go to machine M2; then $L'_2 = B_2 + L_2 - c + \overline{\text{SUM}} - (L_1 + L_2 + B) = B_2 - c + \overline{\text{SUM}} - L_1 - B = -c + \overline{\text{SUM}} - B = \overline{\text{SUM}} - 1 = s$.

Final Case G3. Suppose $L_1 = T_4$ and $L_2 = 0$.

The next and last two jobs are $B = 1$ and $C = \overline{\text{SUM}} - (T_4 + 1) = (T_2 + B_1) - T_4 - 1 = B_1 > 1$, by Lemmas 4.6, 4.2, and 7.4. If any of B or C is assigned to M1, then $L'_1 \geq T_4 + 1 = T_2$ by Lemma 4.2. Thus the lower bound holds. Otherwise both go to M2 and $L'_2 = \overline{\text{SUM}} - T_4 > \overline{\text{SUM}} - B_2 = T_1$ by Lemmas 5.2 and 4.6, and we are done again.

The jobs belong to \mathcal{J}_{OPT} by Lemma 1, because we can assign $B = 1$ to machine M1 and all other jobs to machine M2, which have a total sum of $T_4 + C = T_4 + \overline{\text{SUM}} - (T_4 + 1) = \overline{\text{SUM}} - 1 = (s + 1) - 1 = s$.

3 Lower Bound for Small s

At the end of this section we will prove that $r_1(s)$ is a lower bound on the competitive ratio for small s . Before giving this construction, we consider a number of cases, from which the lower bound can be achieved soon. We start with some further estimations.

3.1 Properties

Lemma 9 *If s is small, then $\frac{2s-1}{s} \leq r \leq \frac{5s+2}{4s+1} \leq \frac{2}{s}$.*

Proof.

- For the first estimation, we see that $\frac{6s+6}{4s+5} - \frac{2s-1}{s} = \frac{5-2s^2}{(4s+5)s} \geq 0$ holds for all $s \leq \frac{\sqrt{10}}{2} \approx 1.5811$, in particular, for small s .
- For the second estimation, we obtain that $\frac{5s+2}{4s+1} - \frac{6s+6}{4s+5} = \frac{-4s^2+3s+4}{(4s+1)(4s+5)} \geq 0$ for $s \leq \frac{3+\sqrt{73}}{8} = q_2$, in particular, for small s .

- For the third estimation, we compute that $\frac{2}{s} - \frac{5s+2}{4s+1} = \frac{-5s^2+6s+2}{s(4s+1)} \geq 0$ for all $s \leq \frac{3+\sqrt{19}}{5} \approx 1.4717$, in particular, for small s .

■

Lemma 10 1. $(T_3 - B_2) + c = b + c \leq B_4$,

2. $c \leq B_4$,

3. $2c \geq B_4$,

4. $2c \geq B_3$,

5. $B_3 = D_4$,

6. $c < B_3$,

7. $B_2 + B_4 < 1$.

Proof.

1. $B_4 - b - c = (4s - 2r - 3rs + 3) - (2rs - 2s - 1) - (s - r - rs + 2) = 5s - r - 4rs + 2 \geq 0$, since $r \leq \frac{5s+2}{4s+1}$ by Lemma 9.
2. This follows from Lemma 10.1, together with $b \geq 0$ from Lemma 7.1.
3. $2c - B_4 = 2(s - r - rs + 2) - (4s - 2r - 3rs + 3) = rs - 2s + 1 \geq 0$, as $r \geq \frac{2s-1}{s}$ from Lemma 9.
4. $2c - B_3 = 2(s - r - rs + 2) - (2s - 2r - rs + 2) = 2 - rs \geq 0$, as $r \leq \frac{2}{s}$ from Lemma 9.
5. $B_3 - D_4 = (2s - 2r - rs + 2) - (3r - 4)(s + 1) = 6s + 6 - r(4s + 5) = 0$, since $r = r_1(s)$ for small s .
6. $B_3 - c = (2s - 2r - rs + 2) - (s - r - rs + 2) = s - r > 0$, as $r < s$ by Lemma 6.1.
7. Since $1 = B_2 + T_3$ by Lemma 4.5, we have to show that $B_4 < T_3$. From Lemma 5.2 and the definitions of T_3 and T_4 it follows that $B_4 < T_4 = r - 1 < s(r - 1) = T_3$.

■

3.2 Subcases

The adversary constructs a sequence of jobs in such way that any assignment strategy of an arbitrary algorithm will lead to one of the ‘‘S’’-cases described below, which make use of the fact that s is small. When the assumptions of these cases are fulfilled, the adversary knows how to define the next jobs, so that the algorithm is trapped and must return a solution having a ratio worse than or equal to $r = r_1(s)$.

Case S1. Assume that $L_1 = B_2$ and $L_2 = B_4 - c$.

Then the next jobs are $B = 1 - (B_4 - c) \geq c - (B_4 - c) = 2c - B_4 \geq 0$ (by Lemma 10.3) and $C = \text{SUM} - (L'_1 + L'_2 + B) = (B_2 + T_1) - (B_2 + 1) = T_1 - 1 = (T_1 - D_1) - c = B_1 - c \geq D_2$, since $B_1 \geq D_2 + c = 1$ holds by Lemma 7.4. Note that $B = 1 - (B_4 - c) = (D_2 + c) - (B_4 - c) = D_2 + 2c - B_4 \geq D_2$ also holds by Lemma 10.3. If any of B or C is assigned to M1, we are done. Indeed, if B is assigned to M1, then $L'_1 = L_1 + B \geq B_2 + D_2 = T_2$, and similarly for C . Otherwise both B and C are assigned to M2, and we are done again: $L'_2 = B_4 - c + B + C = B_4 - c + (1 - (B_4 - c)) + (T_1 - 1) = T_1$.

The set of jobs A, B, C and the previous load of M2 belong to \mathcal{J}_{OPT} (by Lemma 1): assign the previous load of machine M2 and B to machine M1, then its load is $L_2 + B = (B_4 - c) + (1 - (B_4 - c)) = 1$. The remaining jobs A and C go to machine M2, which then has a load of $A + C = B_2 + (B_1 - c) = (s + 1 - r) + (s + 1 + sr) - (s - r - rs + 2) = s$.

Case S2. Suppose $L_1 = 0$ and $L_2 = B_4 - c \geq 0$ (by Lemma 10.2).

Let the next job be $A = B_2$. Suppose A is assigned to M2. Then $L'_1 + L'_2 = B_2 + B_4 - c \leq B_2 + T_4 \leq B_2 + T_3 = 1$ holds (applying $c \geq 0$, $T_4 = r - 1 \leq s(r - 1) = T_3$, and Lemma 4.3). Moreover by Lemma 10.1 we get $L'_2 = B_2 + B_4 - c \geq T_3$, thus case G1 holds for the new loads L'_1, L'_2 , and hence we are done. Otherwise A is assigned to M1. At this moment $L'_1 = B_2$ and $L'_2 = B_4 - c$. Then we are in case S1.

Case S3. Suppose $L_1 = B_4 - c$ and $L_2 = c$.

Let the next job be $A = B_2 - (B_4 - c) = D_3 + c$ by Lemma 4.4. Suppose A is assigned to M2. Then $L'_1 + L'_2 = (B_4 - c) + c + B_2 - (B_4 - c) = B_2 + c \leq B_2 + B_4 \leq 1$ by Lemma 10.7. Moreover $L'_2 = D_3 + 2c \geq D_3 + B_3 = T_3$, since $2c \geq B_3$ holds by Lemma 10.4. Thus case G1 holds for L'_1 and L'_2 , and we are done. Otherwise A is assigned to M1. Then the loads are $L'_1 = B_2$ and $L'_2 = c \leq B_3$ by Lemma 10.6. Hence, case G2 holds for the loads L'_1 and L'_2 , and we are done again.

3.3 The Construction

Consider an arbitrary algorithm to solve the semi-online scheduling problem, where the values s and OPT are known. In the following construction, we take the point of view of an adversary, and try to make the algorithm’s life as hard

as possible. More formally, we will show that the algorithm shall provide a schedule whose competitive ratio is at least $r_1(s)$. Although the whole family of jobs belongs to \mathcal{J}_{OPT} (and the jobs have a total size of $(s+1)\text{OPT}$), the adversary still has enough freedom to force the algorithm to construct an assignment where it ends up with a load L_1 on machine M1 with $L_1 \geq T_2 = r$ or a load L_2 on machine M2 with $L_2 \geq T_1 = rs$.

The adversary decides that the first job is $J_1 = B_4 - c$. This job has a non-negative size by Lemma 10.2. If J_1 goes to M2, then case S2 is satisfied, and we are done (i.e., we trapped the algorithm as explained above). We conclude that J_1 goes to M1.

The second job is $J_2 = c$. If J_2 goes to M2, then case S3 is satisfied, and we are done. We conclude that J_2 goes to M1. At this moment the loads are $L_1 = B_4$ and $L_2 = 0$.

The third job is $J_3 = B_3$. Suppose J_3 goes to M1. Since $L'_1 = B_4 + B_3 = T_4$ holds by Lemma 10.5, we are in case G3, and thus we are done. We conclude that J_3 goes to M2. At this moment the loads are $L_1 = B_4$ and $L_2 = B_3$.

Then the next (and final) job is $J_4 = D_3$. If J_4 goes to M2, then $L'_1 = B_4$ and $L'_2 = T_3$. We estimate that $L'_1 + L'_2 = T_3 + B_4 < T_3 + B_2 = \text{OPT}$, where we applied first Lemma 5.2 and then Lemma 4.5. Thus we showed that we are in case G1, and we are done. We conclude that J_4 goes to M1. At this moment $L_1 = B_4 + D_3 = B_2$ by Lemma 4.4, and $L_2 = B_3$. Now we are in case G2, and we are done.

We remark that the sequence of jobs can be drawn as a decision tree, with the first job at its root node, and all other jobs at the subsequent nodes. A left branch means that the job at a node is assigned to machine M1, and a right branch means that it is assigned to machine M2. Note that this tree has a depth of 6 jobs.

4 Lower bounds for regular and medium s

Here we consider the four cases of s being small regular, small medium, bigger medium, or bigger regular, respectively. We need several further properties regarding the lower bounds.

4.1 Properties

Lemma 11 *If $s \geq \frac{5}{3}$, then*

1. $\max \left\{ \frac{2}{s}, \frac{5s+6}{4s+4}, \frac{6s+5}{6s+1} \right\} \leq r \leq \frac{7s+5}{6s+2},$

2. *if s is small medium, then $\frac{11s+8}{8s+6} \leq r$ also holds.*

Proof.

1. Regarding the lower bounds, applying $\frac{4}{3} < 1.35 < r$ from Lemma 3.1, we get $\frac{4}{3} - \frac{2}{s} = \frac{2(2s-3)}{3s} > 0$ and $\frac{135}{100} - \frac{5s+6}{4s+4} = \frac{2s-3}{20(s+1)} > 0$; both inequalities are true since $s \geq \frac{5}{3}$. Let us see $r \geq \frac{6s+5}{6s+1}$. We have already seen this for regular speeds in Lemma 6.1. For smaller medium s we get $\frac{18s+16}{16s+7} - \frac{6s+5}{6s+1} = \frac{12s^2-8s-19}{(16s+7)(6s+1)} \geq 0$, which is true for $s \geq \frac{3+\sqrt{73}}{8} \approx 1.443$, in particular for $s \geq \frac{5}{3}$. Considering bigger medium s , we get $\frac{8s+7}{3s+10} - \frac{6s+5}{6s+1} = \frac{30s^2-25s-43}{(3s+10)(6s+1)} \geq 0$, which is true for $s \geq \frac{25+\sqrt{5785}}{60} \approx 1.684$, in particular for $s \geq q_4$.

Regarding the upper bound, by Lemma 2 it is enough to show that $r_2(s)$ does not exceed it. We get $\frac{7s+5}{6s+2} - \frac{12s+10}{9s+7} = \frac{-9s^2+10s+15}{54s^2+60s+14} \geq 0$, which is true since $-9s^2+10s+15 \geq 0$ for $\frac{5-4\sqrt{10}}{9} \leq s \leq \frac{5+4\sqrt{10}}{9} \approx 1.961$, in particular, if s is regular or medium.

2. We get $\frac{18s+16}{16s+7} - \frac{11s+8}{8s+6} = \frac{-32s^2+31s+40}{(16s+7)(8s+6)} \geq 0$, which is true, since $-32s^2+31s+40 \geq 0$ for $\frac{31-\sqrt{6081}}{64} \leq s \leq \frac{31+\sqrt{6081}}{64} \approx 1.703$, in particular for $q_3 \leq s \leq q_4$.

■

Lemma 12 1. $s \geq T_2 + c$.

2. $B_2 \leq s - 1$.
3. $T_4 + B_2 - 2c > T_3$, i.e., $T_4 - 2c > T_3 - B_2 = b$, i.e., $T_4 > b + 2c$.
4. $T_4 \geq d$.
5. $T_4 + c \leq B_3 + d$, i.e., $c + (T_4 - B_3) \leq d$, i.e., $c + a \leq d$.
6. $c \leq B_3$.
7. $T_4 + B_2 + c - d < 1 = B_2 + T_3$ (cf. Lemma 4.5), i.e., $T_4 + c < T_3 + d$.
8. $2T_3 + 2B_4 \leq s$.
9. If s is regular, then $3B_4 = T_4 < B_2$.
10. $T_4 \leq 3B_4 \leq B_2$, if s is small medium.
11. $B_3 + B_4 \leq s - 1$.

Proof. We apply Lemma 11, unless stated otherwise.

1. From the definitions of T_2 and c we get $s - T_2 - c = s - r - (s - r - rs + 2) = rs - 2 \geq 0$, which is true, because $r \geq \frac{2}{s}$.
2. As before, we obtain $s - 1 - B_2 = s - 1 - (s + 1 - sr) = rs - 2 \geq 0$.

3. We have from the definitions: $T_4 - b - 2c = (r - 1) - (2rs - 2s - 1) - 2(s - r - rs + 2) = 3r - 4 > 0$, where the last inequality was shown in Lemma 3.1.
4. We get $T_4 - d = (r - 1) - (3r - 7s + 6rs - 6) = 7s + 5 - r(6s + 2) \geq 0$, since $r \leq \frac{7s+5}{6s+2}$.
5. We compute $d - c - a = (3r - 7s + 6rs - 6) - (s - r - rs + 2) - (3r - 2s + rs - 3) = r - 6s + 6rs - 5 \geq 0$, which follows from $r \geq \frac{6s+5}{6s+1}$.
6. Applying Lemmas 12.4 and 12.5, we get $c \leq B_3 + d - T_4 \leq B_3$.
7. This follows from Lemmas 12.5 and 5.1.
8. From the definitions we obtain $s - 2T_3 - 2B_4 = s - 2s(r - 1) - 2(4s - 2r - 3rs + 3) = 4r - 5s + 4rs - 6 \geq 0$, since $r \geq \frac{5s+6}{4s+4}$.
9. From the definitions we have $3B_4 - T_4 = 3(4s - 2r - 3rs + 3) - (r - 1) = 12s + 10 - r(9s + 7) = 0$, since $r = r_2(s)$. Moreover, $T_4 < B_2$ by Lemma 5.2.
10. For small medium s , it holds that $3B_4 - T_4 = 12s + 10 - r(9s + 7) \geq 0$, since $r = r_3(s) \leq r_2(s)$ by Lemma 2.3. Moreover, $B_2 - 3B_4 = (s + 1 - sr) - 3(4s - 2r - 3rs + 3) = 6r - 11s + 8rs - 8 \geq 0$, as $r \geq \frac{11s+8}{8s+6}$.
11. We estimate $s - 1 - B_3 - B_4 = s - 1 - (2s - 2r - rs + 2) - (4s - 2r - 3rs + 3) = 4r - 5s + 4rs - 6 \geq 0$, since $r \geq \frac{5s+6}{4s+4}$.

■

In the next lemma we consider only the cases where s is smaller regular or smaller medium. Let $t = \frac{\sqrt{4633+23}}{54} \approx 1.6864$. Recall that $q_3 = \frac{13+\sqrt{1429}}{30} \approx 1.6934$.

- Lemma 13**
1. $T_4 + c \geq b + 2d$ holds if s is smaller regular and $s \leq t \approx 1.6864$,
 2. $2B_2 + c \geq T_3 + 2d$ holds if s is smaller regular and $t \leq s \leq q_3$,
 3. $2B_2 + c = T_3 + 2d$ holds if s is smaller medium,
 4. $c + B_2 \leq B_3 + d$ holds if s is smaller regular and $s \geq t$,
 5. $c + B_2 \leq B_3 + d$ holds if s is smaller medium,
 6. $2B_2 + c - d \leq 1 = B_2 + T_3$, i.e. $B_2 + c \leq T_3 + d$, if s is smaller regular and $s \geq t$, or if s is smaller medium.

Proof.

1. $T_4 + c - (b + 2d) = (r - 1) + (s - r - rs + 2) - (2rs - 2s - 1) - 2(3r - 7s + 6rs - 6) = 17s + 14 - r(6 + 15s) = 17s + 14 - \frac{12s+10}{9s+7}(6 + 15s) = \frac{38+23s-27s^2}{9s+7} \geq 0$, which holds because $38 + 23s - 27s^2 \geq 0$ if and only if $\frac{23-\sqrt{4633}}{54} \leq s \leq \frac{23+\sqrt{4633}}{54}$.

2. $2B_2 + c - 2d - T_3 = 2(s + 1 - sr) + (s - r - rs + 2) - 2(3r - 7s + 6rs - 6) - s(r - 1) = 18s + 16 - r(7 + 16s) = 18s + 16 - (7 + 16s)\frac{12s+10}{9s+7} = \frac{2(21+13s-15s^2)}{9s+7} \geq 0$, which holds because $21 + 13s - 15s^2 \geq 0$ if and only if $\frac{13-\sqrt{1429}}{30} \leq s \leq \frac{13+\sqrt{1429}}{30}$.
3. $2B_2 + c - 2d - T_3 = 18s + 16 - (7 + 16s)r = 18s + 16 - (7 + 16s)\frac{18s+16}{16s+7} = 0$.
4. $B_3 + d - c - B_2 = (2s - 2r - rs + 2) + (3r - 7s + 6rs - 6) - (s - r - rs + 2) - (s + 1 - sr) = (7s + 2)r - 7s - 7 = (7s + 2)\frac{12s+10}{9s+7} - 7s - 7 = \frac{21s^2 - 18s - 29}{9s+7} \geq 0$, since $21s^2 - 18s - 29 \geq 0$ if and only if $s \leq \frac{9-\sqrt{690}}{21}$ or $s \geq \frac{9+\sqrt{690}}{21} \approx 1.6794$.
5. $B_3 + d - c - B_2 = (7s + 2)r - 7s - 7 = \frac{(7s+2)(18s+16)}{16s+7} - 7s - 7 = \frac{14s^2 - 13s - 17}{16s+7} \geq 0$, since $14s^2 - 13s - 17 \geq 0$ if and only if $s \leq \frac{13-\sqrt{1121}}{28}$ or $s \geq \frac{13+\sqrt{1121}}{28} \approx 1.660$.
6. This follows from Lemmas 13.4 and 13.5 using Lemma 4.5.

■

In the next lemma we consider only bigger regular s .

Lemma 14 *Let s be bigger regular. Then*

1. $c + B_2 + T_4 - 2e \geq T_3$, i.e., $c + T_4 - 2e \geq T_3 - B_2 = b$.
2. $c + T_4 \leq B_3 + e$, i.e., $c + (T_4 - B_3) = c + a \leq e$.
3. $c + B_2 + T_4 - e < 1 = B_2 + T_3$, i.e., $c + T_4 < T_3 + e$.
4. $B_4 \geq d$.

Proof.

1. We derive that $c + T_4 - 2e - b = (s - r - rs + 2) + (r - 1) - 2(\frac{1}{2}r - 3s + 3rs - \frac{5}{2}) - (2rs - 2s - 1) = 9s + 7 - r(9s + 1) \geq 0$. The last estimation is true if $\frac{9s+7}{9s+1} - \frac{12s+10}{9s+7} = \frac{3(-9s^2+8s+13)}{(9s+1)(9s+7)} \geq 0$, which holds since $-9s^2 + 8s + 13 \geq 0$ if and only if $\frac{4-\sqrt{133}}{9} \leq s \leq \frac{4+\sqrt{133}}{9} = q_7 \approx 1.7258$, in particular, for bigger regular s .
2. $e - c - a = (\frac{1}{2}r - 3s + 3rs - \frac{5}{2}) - (s - r - rs + 2) - (3r - 2s + rs - 3) = 3rs - 2s - \frac{3}{2}r - \frac{3}{2} \geq 0$, which is true for $r \geq \frac{4s+3}{6s-3}$. Hence we need to verify that $\frac{12s+10}{9s+7} - \frac{4s+3}{6s-3} = \frac{36s^2-31s-51}{3(9s+7)(2s-1)} \geq 0$, which holds for all $s \geq \frac{31+\sqrt{8305}}{72} = q_5 \approx 1.6963$ (and some negative values for s , which we can ignore).
3. This follows from Lemmas 14.2 and 5.1: $c + T_4 \leq B_3 + e < T_3 + e$.

4. Using the definitions of B_4 and d it is to show that $4s - 2r - 3rs + 3 \geq 3r - 7s + 6rs - 6$, which is equivalent to $-9rs - 5r + 11s + 9 \geq 0$ or in other form $r \leq \frac{11s+9}{9s+5}$. Taking into account that $r = r_2(s) = \frac{12s+10}{9s+7}$, we arrive at $-9s^2 + 8s + 13 \geq 0$, which is in particular true for bigger regular values of s (as we have seen in Lemma 14.1).

■

Now we consider the case of bigger medium speeds.

Lemma 15 *If s is bigger medium, then*

1. $4c + 4a = B_4$,
2. $b \leq 8c + 7a \leq B_3$.

Proof.

$$1. \frac{B_4}{4} - c - a = \frac{4s-2r-3rs+3}{4} - (s - r - rs + 2) - (3r - 2s + rs - 3) = -\frac{r(3s+10)-8s-7}{4} = 0.$$

$$2. \text{ Left inequality: } 8c + 7a - b = 8(s - r - rs + 2) + 7(3r - 2s + rs - 3) - (2rs - 2s - 1) = r(13 - 3s) - 4s - 4 = \frac{8s+7}{3s+10}(13 - 3s) - 4s - 4 = \frac{31s+51-36s^2}{3s+10} \geq 0.$$

The inequality is satisfied since $31s + 51 - 36s^2 \geq 0$ holds if and only if $\frac{31-\sqrt{8305}}{72} \leq s \leq \frac{31+\sqrt{8305}}{72} = q_5$.

$$\text{Right inequality: } B_3 - 8c - 7a = (2s - 2r - rs + 2) - 8(s - r - rs + 2) - 7(3r - 2s + rs - 3) = 8s - 15r + 7 = 15\left(\frac{8s+7}{15} - r\right) = 15\left(\frac{8s+7}{15} - \frac{8s+7}{3s+10}\right) = \frac{(8s+7)(3s-5)}{15(3s+10)} \geq 0, \text{ which is true for } s \geq \frac{5}{3}.$$

■

4.2 Subcases

For s being regular or medium (“RM”), we consider several further situations, from which the lower bound can be achieved directly, or that leads to other general cases we dealt with before.

Case RM1. Suppose $T_4 \leq L_1 \leq B_2$ and $0 \leq L_1 + L_2 \leq s - 1$.

Note that we have $s - 1 \geq B_2$ by Lemma 12.2. The next and last two jobs are $B = 1$ and $C = \overline{\text{SUM}} - (L_1 + L_2 + B) \geq s + 1 - s = 1 = B$. If any of B and C is assigned to M1, then $L'_1 \geq T_4 + 1 = T_2$, thus the lower bound holds; otherwise both go to M2 and $L'_2 \geq \overline{\text{SUM}} - B_2 = T_1$, and we are done again.

The set of jobs belongs to \mathcal{J}_{OPT} by Lemma 1: assign B to M1 and the remaining jobs to M2, which has a load of $L_1 + L_2 + C = L_1 + L_2 + \overline{\text{SUM}} - (L_1 + L_2 + B) = \overline{\text{SUM}} - B = s + 1 - 1 = s$.

Case RM2. Suppose $L_1 = c$ and $L_2 = 0$.

Let the next job be $A = T_4 - c$. This is nonnegative by Lemma 12.3, using that $b > 0$ from Lemma 7.1. Suppose A is assigned to M1. At this time $L'_1 = T_4$ and $L_2 = 0$, thus case G3 holds and we are done. Now suppose A goes to M2; then let the next job be $B = B_2 - c$. This is nonnegative, as $c \leq T_4$ (which we already observed) and $T_4 \leq B_2$. If B goes to M2, then the load of M2 will be $L'_2 = A + B = T_4 + B_2 - 2c$. This is at least T_3 by Lemma 12.3. Moreover, using $T_3 = sT_4 > T_4$ (by the definitions of T_3 and T_4) and $c \geq 0$ (by Lemma 7.1), we have $L'_1 + L'_2 = T_4 + B_2 - c < T_3 + B_2 = 1$. Thus we are in case G1, and we are done. Otherwise B goes to M1. At this moment the loads are $L'_1 = B_2$ and $L'_2 = A = T_4 - c \leq B_3$, since $T_4 - B_3 = a \leq c$ by the definition of a and Lemma 7.2. Hence we are in case G2, and we are done.

Case RM3. Suppose $L_1 = T_4$ and $L_2 = c$.

Applying $T_2 - T_4 = 1$ by Lemma 4.2, we get $L_1 + L_2 + 1 = T_4 + 1 + c = T_2 + c \leq s$, by Lemma 12.1, hence $L_1 + L_2 \leq s - 1$. So we are in case RM1, and thus we are done.

Case RM4. Suppose $L_1 = d$ and $L_2 = c$, and s is smaller regular or smaller medium.

1. Assume that $\frac{5}{3} \leq s \leq t \approx 1.6864$.

Note that $L_1 = d > 0$. Let the next job be $A = T_4 - d$. This is nonnegative by Lemma 12.4. If A is assigned to M1, we meet the prerequisites of case RM3, and we are done. Thus suppose A goes to M2. Let the next job be $B = B_2 - d$. (This is positive as $B_2 > T_4$.) If B goes to M2, then $L'_2 = c + A + B$. Moreover, $L'_1 + L'_2 = d + c + (T_4 - d) + (B_2 - d) = T_4 + B_2 + c - d < 1$, applying Lemma 12.7. We state that $L'_2 = T_4 + B_2 + c - 2d \geq T_3$ holds in the considered interval. Indeed, since $b = T_3 - B_2$, it suffices to see that $T_4 + c \geq b + 2d$, which holds by Lemma 13.1. Thus we are in case G1, and we are done. Otherwise B goes to M1. At this moment the loads are $L'_1 = B_2$ and $L'_2 = L_2 + A = c + T_4 - d \leq B_3$ by Lemma 12.5. Thus this is case G2, and we are done.

2. Assume that $t \leq s \leq q_3$ or s is smaller medium.

Note that $0 < L_1 = d \leq T_4$ by Lemma 12.4. Let the next job be $A = B_2 - d$, which is positive, since $B_2 > T_4$ by Lemma 5.2. If A is assigned to M1, then the new load on this machine is $L'_1 = d + B_2 - d = B_2$, while $L'_2 = L_2 = c$. By Lemma 12.6 we get $c \leq B_3$, thus we are in case G2, and we are done. Thus suppose A goes to M2. Then, let the next job be $B = B_2 - d$. If B goes to M2, then the load of M2 will be $L'_2 = c + A + B = c + 2(B_2 - d)$. Then $L'_1 + L'_2 = 2B_2 + c - d$. This is at most 1 by Lemma 13.6. Moreover, $L'_2 = 2B_2 + c - 2d \geq T_3$, by Lemmas 13.2 and 13.3. Thus we are in case

G1, and we are done. Otherwise, B goes to M1. In this moment, the loads are $L'_1 = d + (B_2 - d) = B_2$ and $L'_2 = L_2 + A = c + B_2 - d \leq B_3$, by Lemmas 13.4 and 13.5. We are in case G2, and we are done.

Case RM5. Suppose $L_1 = B_4$, and $L_2 = b - B_4$, and s is small medium.

Let the next job be $A = 2B_4$. Suppose A is assigned to M1. At this time $L'_1 = 3B_4$ and $L_2 = b - B_4$. Note that $T_4 \leq 3B_4 \leq B_2$ by Lemma 12.10. Using Lemma 4.5, the definition of b , and Lemma 12.8, we get $L'_1 + L'_2 + 1 = 3B_4 + (b - B_4) + (T_3 + B_2) = 2B_4 + (T_3 - B_2) + (T_3 + B_2) = 2T_3 + 2B_4 \leq s$. Hence we are in case RM1, and we are done. We conclude that A is assigned to M2. Let the next job be $B = B_2 - B_4$ (this is positive, since $B_2 > B_4$). If B goes to M2, then

$$L'_2 = L_2 + A + B = (b - B_4) + 2B_4 + (B_2 - B_4) = (T_3 - B_2) + B_2 = T_3$$

by the definition of b . Therefore $L'_1 + L'_2 = B_4 + T_3 < B_2 + T_3 = 1$, by Lemmas 5.2 and 4.5. Since L'_2 equals T_3 , we are in case G1, and we are done. Otherwise B goes to M1. At this moment the loads are $L'_1 = B_2$ and

$$\begin{aligned} L'_2 &= L_2 + A = (b - B_4) + 2B_4 = (T_3 - B_2) + B_4 \\ &= (B_3 + D_3) - B_2 + B_4 = B_3 + (B_2 - B_4) - B_2 + B_4 = B_3, \end{aligned}$$

using $T_3 = B_3 + D_3$ and $D_3 = B_2 - B_4$ (by Lemma 4.4). Hence we are in case G2, and we are done.

Case RM6. Suppose $L_1 = e$ and $L_2 = c$, and s is bigger regular.

Let the next job be $A = T_4 - e$. This job is nonnegative by Lemma 12.4, $A \geq d - e$, and $d = e + f \geq e$ by Lemma 7.1. Suppose A is assigned to M1. Then $L'_1 = T_4$ and $L'_2 = c$, thus case RM3 holds, and so we are done. Otherwise, A goes to M2. Let the next job be $B = B_2 - e$. (This job is positive, since $B_2 > T_4$ from Lemma 5.2, and by the observation $T_4 \geq e$ above.) If B goes to M2, then the load of M2 will be $L'_2 = c + (T_4 - e) + (B_2 - e) = c + B_2 + T_4 - 2e \geq T_3$ by Lemma 14.1. Then $L'_1 + L'_2 = c + B_2 + T_4 - e$. This is smaller than 1 by Lemma 14.3. Thus we are in case G1, and we are done. Otherwise B goes to M1. At this moment, the loads are $L'_1 = B_2$ and $L'_2 = c + T_4 - e \leq B_3$ by Lemma 14.2. Hence we are in case G2, and we are done again.

Case RM7. Suppose $L_1 = L_2 + a \leq B_4$.

Let the next job be $A = T_4 - L_1$. This is positive as $L_1 \leq B_4$ and $B_4 < T_4$ (by Lemma 5.2). Suppose A is assigned to M1. Since $L'_1 + L'_2 \leq T_4 + B_4 - a = T_4 + B_4 - (T_4 - B_3) = B_3 + B_4 \leq s - 1$ holds by the definition of a and Lemma 12.11, we are in case RM1, and we are done. Now suppose A goes to M2. Then let the next job be $B = B_2 - L_1$, which is positive by Lemma 5.2. If B goes to M2, then the load of M2 will be $L'_2 =$

$(L_1 - a) + (T_4 - L_1) + (B_2 - L_1) = B_2 + T_4 - a - L_1 \geq B_2 + T_4 - (T_4 - B_3) - B_4 = (B_2 - B_4) + B_3 = D_3 + B_3 = T_3$, by the definition of a and Lemma 4.4. On the other hand, $L'_1 + L'_2 = B_2 + T_4 - a = B_2 + T_4 - (T_4 - B_3) = B_2 + B_3 < B_2 + T_3 = 1$, by the definition of a and Lemma 5.2. Thus we are in case G1, and we are done. Otherwise B goes to M1. At this moment the loads are $L'_1 = B_2$ and $L'_2 = (L_1 - a) + (T_4 - L_1) = T_4 - a = T_4 - (T_4 - B_3) = B_3$ by the definition of a . Thus we meet case G2, and we are done again.

Case RM8. Suppose $L_1 = 0$, and $b \leq L_2 \leq B_3$.

Let the next job be $A = B_2$. If A goes to M2, then the increased load of M2 will be $L'_2 = L_2 + B_2 \geq b + B_2 = (T_3 - B_2) + B_2 = T_3$ (by the definition of b), and $L'_1 + L'_2 \leq B_2 + B_3 < B_2 + T_3 = 1$ (by Lemma 5.1 and Lemma 4.5), thus case G1 is satisfied, and we are done. Otherwise, A is assigned to M1. We then meet case G2, and we are also done.

4.3 The Construction

Similarly to the construction for small s in Section 3.3, we construct a sequence of jobs such that any semi-online algorithm knowing s and OPT will assign in such way that the competitive ratio is at least $r(s)$, where s is between $\frac{5}{3}$ and q_7 . Again, the sequence of jobs belongs to \mathcal{J}_{OPT} , and the total size of the jobs is $\overline{\text{SUM}} = s + 1$.

First, the adversary chooses the job $J_1 = c$. If J_1 goes to M1, then case RM2 is satisfied, and we are done. We conclude that J_1 goes to M2.

We divide the further construction into two main cases, depending on the value of s .

Case 1: s is smaller regular, smaller medium, or bigger regular.

Case 1.1: s is smaller regular or smaller medium.

The second job is $J_2 = d$. If J_2 goes to M1, then case RM4 is satisfied, and we are done. Thus we conclude that J_2 goes to M2. At this point $L_1 = 0$ and $L_2 = d + c = (d - e) + (c + e) = f + g = b - B_4$. We continue the construction after case 1.2.

Case 1.2: s is bigger regular.

The second job is $J_{21} = e$. If J_{21} goes to M1, then the assumption of case RM6 is satisfied, and we are done. We conclude that J_{21} goes to M2. The next job is $J_{22} = f$. Suppose J_{22} goes to M1. Then $L_1 = f$ and $L_2 = J_1 + J_{21} = c + e = g$. From Lemmas 14.4 and 7.1 it follows that $B_4 \geq d = e + f \geq f$. Since we also have that $f - g = a$, we are altogether in case RM7, thus we are done. We conclude that J_{22} goes to M2. At this point $L_1 = 0$ and $L_2 = c + e + f = f + g = b - B_4$.

Now we join the treatments of these subcases, Case 1.1 and Case 1.2, and finish the construction. In both subcases now $L_1 = 0$ and $L_2 = b - B_4$. Then comes $J_3 = B_4$. Suppose J_3 goes to M1. Then $L_1 = B_4$ and $L_2 =$

$b - B_4$. If s is small medium, then we are in case RM5, and we are done. Otherwise, if s is small regular or bigger regular, then we claim that we are in case RM7, for which we have to show $L_2 + a = L_1 \leq B_4$. The inequality on the right follows already from the construction, and it remains to show $L_1 = L_2 + a$. Thus by the definitions of a and b and Lemma 4.4, we obtain $L_1 - L_2 - a = B_4 - (b - B_4) - a = 2B_4 - (T_3 - B_2) - (T_4 - B_3) = 2B_4 + (B_2 - D_3) - T_4 = 2B_4 + B_4 - T_4 = 3B_4 - T_4 = 0$, where the last equality was shown in Lemma 12.9. Thus we enter case RM7, and we are done.

We conclude that J_3 goes to M2, and at this moment $L_1 = 0, L_2 = b$. We claim that we are in case RM8 then, for which we need to show that $L_2 \leq B_3$. From Lemma 5.2 we know that $B_4 > 0$. Hence we obtain from Lemma 4.4 and the definition of D_3 that $B_2 = B_4 + D_3 > D_3 = T_3 - B_3$. We conclude that $B_3 > T_3 - B_2 = b$, by the definition of b .

Case 2: s is bigger medium.

The second job is $J_{21} = c + a$. If J_{21} goes to M1, then the assumptions $L_1 = L_2 + a \leq B_4$ of case RM7 are satisfied, because $B_4 = 4c + 4a > c + a > 0$ by Lemmas 15.1 and 5.2, and we are done. We conclude that J_{21} goes to M2. Then comes $J_{22} = 2c + 2a$. If J_{22} goes to M1, then $L_1 = 2c + 2a$ and $L_2 = J_1 + J_{21} = 2c + a$. Thus we are again in case RM7 (by repeating the previous arguments), and we are done. We conclude that J_{22} goes to M2. At this point $L_1 = 0$ and $L_2 = 4c + 3a$. Then comes $J_{23} = 4c + 4a$. If J_{23} goes to M1, then $L_1 = 4c + 4a$ and $L_2 = 4c + 3a$. Thus case RM7 is applicable again (by applying Lemma 15.1), and we are done. We conclude that J_{23} goes to M2. At this moment $L_1 = 0$ and $L_2 = 8c + 7a$. By Lemma 15.2 we know that $b \leq 8c + 7a \leq B_3$. Hence we are in case RM8, and we are done.

Again, it is possible to sketch the above assignment steps in a decision tree, as explained at the end of Section 3.3. The depth of this tree depends on the value of s . For smaller regular and smaller medium s , we have a depth of 7 jobs, and for bigger medium and bigger regular s , we have a depth of 8 jobs.

5 Main Theorem

The following theorem summarizes the work done above.

Theorem 16 *The function $r(s)$ (defined in Section 2.1) is a lower bound on the optimal competitive ratio for the two uniform machine semi-online scheduling problem with known optimal offline objective function value.*

Together with the algorithms from Ng et al. [20], we then obtain:

Corollary 17 *The lower bound given by $r(s)$ is tight for $[q_1, q_2] \approx [1.3956, 1.443]$, moreover for $[\frac{5}{3}, q_3] \approx [1.6666, 1.6934]$, and also for $[q_5, q_6] \approx [1.6963, 1.7103]$.*

6 Conclusions and Outlook

Starting with the work of Epstein [11] on the semi-online two uniform machines scheduling problem with known optimum, researchers have continued to close the gap between lower and upper bounds. As one can deduce from Figure 1, this goal has been achieved for large portions of the line $[1, \infty)$. We contributed to this ultimate goal by giving new lower bounds and thus showing that some already existing algorithms (of Ng et al. [20]) are in fact best possible, so our bounds are tight. Our new results give insight into the difficulty of the problem: Why is it so hard to give the tight competitive ratio for this model? In part, an answer lies in the fact that a single algebraic function cannot describe the tight lower bound. From what is known by now, at least six different piecewise-defined algebraic functions are necessary. And still, the question of the optimal competitive ratio is open on certain parts of the “right” interval, namely in $(q_2, \frac{5}{3})$, (q_3, q_5) , and $(q_6, \sqrt{3})$. The latter two we called the *narrow interval* and *wide interval*, respectively. Regarding the wide interval, in the continuation [7] of this paper we prove that our lower bound presented here or the “old” lower bound due to Epstein [11] is in fact tight.

Acknowledgements. Krzysztof Węsek’s work was partially supported by the European Union in the framework of European Social Fund through the Warsaw University of Technology Development Programme, realized by Center for Advanced Studies. Furthermore, Węsek’s work was conducted as a guest researcher at the Helmut Schmidt University.

References

- [1] S. Albers. Better bounds for online scheduling. *SIAM Journal of Computing*, 29:459 – 473, 1999.
- [2] E. Angelelli, M. G. Speranza, and Z. Tuza. Semi-online scheduling on two uniform processors. *Theoretical Computer Science*, 393:211 – 219, 2008.
- [3] E. Angelelli, M. G. Speranza, and Zs. Tuza. Semi on-line scheduling on three processors with known sum of the tasks. *Journal of Scheduling*, 10:263 – 269, 2007.
- [4] Y. Azar and O. Regev. On-line bin-stretching. *Theoretical Computer Science*, 268(1):17 – 41, 2001.
- [5] P. Berman, M. Charikar, and M. Karpinski. On-line load balancing for related machines. *Journal of Algorithms*, 35:108 – 121, 2000.
- [6] T. C. E. Cheng, H. Kellerer, and V. Kotov. Semi-on-line multi-processor scheduling with given total processing time. *Theoretical Computer Science*, 337:134 – 146, 2005.

- [7] Gy. Dósa, A. Fügenschuh, Z. Tan, Zs. Tuza, and K. Węsek. Semi-Online Scheduling on Two Uniform Machines with Known Optimum Part II: Tight Upper Bounds. Technical report, Applied Mathematics and Optimization Series AMOS#28, Helmut Schmidt University/University of the Federal Armed Forces, Hamburg, Germany, 2015.
- [8] Gy. Dósa, M. G. Speranza, and Zs. Tuza. Two uniform machines with nearly equal speeds: unified approach to known sum and known optimum in semi on-line scheduling. *Journal of Combinatorial Optimization*, 21:458 – 480, 2011.
- [9] T. Ebenlendr and J. Sgall. A lower bound on deterministic online algorithms for scheduling unrelated machines without preemption. In *Proceeding of the 9th Workshop on Approximation and Online Algorithms*, Lecture Notes in Computer Science, pages 102 – 108, 2007.
- [10] T. Ebenlendr and J. Sgall. Semi-Online Preemptive Scheduling: One Algorithm for All Variants. *Theory of Computer Systems*, 48:577 – 613, 2011.
- [11] L. Epstein. Bin stretching revisited. *Acta Informatica*, 39:97 – 117, 2003.
- [12] U. Faigle, W. Kern, and G. Turán. On the performance of on-line algorithm for particular problem. *Acta cybernetica*, 9:107 – 119, 1989.
- [13] R. Fleischer and M. Wahl. Online scheduling revisited. *Journal of Scheduling*, 3:343 – 353, 2000.
- [14] T. Gormley, N. Reingold, E. Torng, and J. Westbrook. Generating adversaries for request-answer games. In *Proceeding of the 11th ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York/Society for Industrial and Applied Mathematics, Philadelphia, 2000.
- [15] R. L. Graham. Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, 45:1563 – 1581, 1966.
- [16] R. L. Graham, E. L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5, 1979.
- [17] H. Kellerer, V. Kotov, M. G. Speranza, and Zs. Tuza. Semi on-line algorithms for the partition problem. *Operations Research Letters*, 21:235 – 242, 1997.
- [18] K. Lee and K. Lim. Semi-online scheduling problems on a small number of machines. *Journal of Scheduling*, 16:461 – 477, 2013.
- [19] X. Lu and Z. Liu. Semi-online scheduling problems on two uniform machines under a grade of service provision. *Theoretical Computer Science*, 489 – 490:58 – 66, 2013.

- [20] C. T. Ng, Z. Tan, Y. He, and T. C. E. Cheng. Two semi-online scheduling problems on two uniform machines. *Theoretical Computer Science*, 410(8 – 10):776 – 792, 2009.
- [21] Z. Tan and A. Zhang. Online and Semi-online Scheduling. In P. M. Pardalos et al., editor, *Handbook of Combinatorial Optimization*. Springer Science+Business Media New York, 2013.

