Journal of Combinatorial Optimization
https://doi.org/10.1007/s10878-020-00633-w

# Online machine minimization with lookahead

**Cong Chen[1] · Huili Zhang[2] · Yinfeng Xu[2]**



**Abstract**

This paper studies the online machine minimization problem, where the jobs have real release times, uniform processing times and a common deadline. We investigate how the lookahead ability improves the performance of online algorithms. Two lookahead models are studied, that is, the *additive lookahead* and the *multiplicative lookahead*. At any time $t$, the online algorithm knows all the jobs to be released before time $t + L$ (or $\beta \cdot t$) in the additive (or multiplicative) lookahead model. We propose a $\frac{e}{\alpha(e-1)+1}$-competitive online algorithm with the additive lookahead, where $\alpha = \frac{L}{T} \leq 1$ and $T$ is the common deadline of the jobs. For the multiplicative lookahead, we provide an online algorithm with a competitive ratio of $\frac{\beta e}{(\beta-1)e+1}$, where $\beta \geq 1$. Lower bounds are also provided for both of the two models, which show that our algorithms are optimal for two extreme cases, that is, $\alpha = 0$ (or $\beta = 1$) and $\alpha = 1$ (or $\beta \to \infty$), and remain a small gap for the cases in between. Particularly, for $\alpha = 0$ (or $\beta = 1$), the competitive ratio is $e$, which corresponds to the problem without lookahead. For $\alpha = 1$ (or $\beta \to \infty$), the competitive ratio is 1, which corresponds to the offline version (with full information).

**Keywords** Online machine minimization · Online scheduling · Online algorithm · Lookahead

## 1 Introduction

We consider a classical scheduling problem called *machine minimization*, in which the goal is to find a schedule that uses a minimum number of machines to complete all the jobs before the deadlines. The problem is important because in many real world applications, the activation cost of a machine is dominant, and thus a crucial objective is to reduce the number of activated machines. For example, in the train rostering

✉ Huili Zhang
zhang.huilims@xjtu.edu.cn

1 School of Business Administration, South China University of Technology, Guangzhou, China

2 School of Management, Xi'an Jiaotong University, Xi'an, China

Published online: 09 August 2020

problem (Eidenbenz et al. 2003), saving trains is an important financial objective for train operators. In some manufacturing industries (e.g. pharmaceutical factories and steel-making factories), the activation cost of a machine is extremely high, and hence controlling the total number of activated machines is significantly important. The problem also plays an important role in some door-to-door service arrangement, where the goal is to fulfill all the customers by a minimum number of servers.

The machine minimization problem is to schedule a set of jobs on the minimum number of machines required. Each job $J_i$ has a release time $r_i$, a processing time $p_i$, and must be completed by its deadline $d_i$. Since the jobs/orders are often revealed online over time in practice, we study the online setting of the machine minimization problem, where each job is known to the algorithm at its release time $r_i$. Thus, the algorithm has to decide whether to activate a new machine depending only on the information of the jobs that have been released so far, without any knowledge of the future jobs. The goal is to design some online algorithm that minimizes the number of activated machines to ensure that all the jobs are completed by their deadlines.

Studies could be classified into two categories depending on whether the job preemption is allowed. A summary of related results are shown in Table 1, where "Pmtn" defines the preemption is allowed or not; "PT" defines the processing times of jobs are real number ($\mathbb{R}$), non-negative integers ($\mathbb{N}$) or unit value (1); "DDL" defines the deadlines of jobs are $\mathbb{R}$, $\mathbb{N}$ or uniform ($T$); "TL" defines the time line is contiguous (C) or discretized (D); "LA" defines whether the lookahead ability is considered; "UB" and "LB" define the upper and lower bounds.

When preemption is allowed, any job in process may be paused and resumed later, possibly on a different machine. The preemptive version of online machine minimization problem has been investigated extensively. Phillips et al. (2002) showed that there is an $O(\log \frac{p_{\max}}{p_{\min}})$-competitive online algorithm, where $p_{\max}$ and $p_{\min}$ are the maximum and minimum processing times of jobs, respectively. However, the competitive ratio is far from optimal compared with a lower bound of $5/4$ they provided. Since then, no significant improvement had been made until nearly two decades later. Chen et al. (2018) improved the competitive ratio to $O(\log m)$, where $m$ is the optimal number of machines in the offline setting (all the jobs are known in advance). Based on this novel work, the competitive ratio was further improved to $O(\frac{\log m}{\log\log m})$ (Azar and Cohen 2018) and to $O(\log\log m)$ (Im et al. 2017).

The non-preemptive version of online machine minimization problem is more challenging. In fact, Saha (2013) showed that no algorithm can achieve a competitive ratio better than $\Omega(\log \frac{p_{\max}}{p_{\min}})$, which is unbounded if $\frac{p_{\max}}{p_{\min}}$ is unbounded. Thus, more research focused on special cases of the problem, such as jobs with uniform processing times and with a common deadline. Kao et al. (2012) provided a 5.2-competitive algorithm and a lower bound of 2.09 for the online machine minimization problem with uniform job processing times. Then Devanur et al. (2014) improved the competitive ratio to $e$, and showed that no deterministic algorithm for this problem has a competitive ratio less than $e$ (the same results were also implied by the context of energy minimization (Bansal et al. 2007)). Devanur et al. (2014) also gave a 16-competitive algorithm for the special case of jobs with equal deadlines.

**Table 1** A summary of related results

| Authors | Pmtn | PT | DDL | TL | LA | UB | LB |
|---|---|---|---|---|---|---|---|
| Phillips et al. (2002) | Yes | $\mathbb{R}$ | $\mathbb{R}$ | C | 0 | $O\left(\log \frac{p_{\max}}{p_{\min}}\right)$ | $\frac{5}{4}$ |
| Chen et al. (2018) | Yes | $\mathbb{N}$ | $\mathbb{N}$ | C | 0 | $O(\log m)$ | – |
| Azar and Cohen (2018) | Yes | $\mathbb{N}$ | $\mathbb{N}$ | C | 0 | $O\left(\frac{\log m}{\log\log m}\right)$ | – |
| Im et al. (2017) | Yes | $\mathbb{N}$ | $\mathbb{N}$ | C | 0 | $O(\log\log m)$ | – |
| Saha (2013) | No | $\mathbb{R}$ | $\mathbb{R}$ | C | 0 | $O\left(\log \frac{p_{\max}}{p_{\min}}\right)$ | $\Omega\left(\log \frac{p_{\max}}{p_{\min}}\right)$ |
| Kao et al. (2012) | No | 1 | $\mathbb{N}$ | D | 0 | 5.2 | 2.09 |
| Devanur et al. (2014) | No | 1 | $\mathbb{N}$ | D | 0 | $e$ | $e$ |
| Devanur et al. (2014) | No | $\mathbb{N}$ | $T$ | D | 0 | 16 | – |
| Our work | No | 1 | $T$ | C | $t + L$ | $\frac{e}{\frac{L}{T}(e-1)+1}$ | See Theorem 6 |
| Our work | No | 1 | $T$ | C | $\beta \cdot t$ | $\frac{\beta e}{(\beta-1)e+1}$ | See Theorem 7 |

However, most of the previous results for the non-preemptive online machine minimization problem have a common assumption that the release times and the deadlines are all non-negative integers, that is, $r_i, d_i \in \mathbb{N}$. We will show that (in Sect. 2) without the assumption, any no-waiting online algorithm has an unbounded competitive ratio, where no-waiting algorithms will always keep the machines busy if there is any unprocessed job. This implies all the above algorithms have no bounded guarantees for the problem with real release times and deadlines (i.e., $r_i, d_i \in \mathbb{R}^+$), even for the special case of uniform job processing times. Therefore, we focus on the non-preemptive problem where the jobs have real release times and deadlines, but with an additional assumption that the deadlines are equal.

Since algorithms may also resort to a limited preview on the future information (i.e., the lookahead) by taking advantage of the technological advances in real world, we investigate how the lookahead ability improves the performance of online algorithms. Usually the lookahead ability is modeled as a limited length of time to be foreseen. An online algorithm is said to have a lookahead ability of $L \geq 0$, if at any time $t$ the algorithm has the information of all the jobs (to be) released before time $t + L$. The lookahead ability gives an online algorithm some extra power to improve its competitive ratio. The case with lookahead has been addressed in several problems, such as routing and transportation (Allulli et al. 2008; Jaillet and Wagner 2008; Zhang et al. 2016), scheduling (Li et al. 2009; Mandelbaum and Shabtay 2011; Zheng et al. 2013, 2008), etc. Particularly, Zheng et al. (2013) studied the online weighted interval scheduling problem on a single machine, where the limited lookahead was considered. They investigated how the lookahead ability improves the competitive ratio of the online algorithm. The interval scheduling problem is similar to the setting of integer release times and deadlines in machine minimization problem. However, the objective is quite different, since one is to maximize the total weight of the completed jobs and one is to minimize the number of machines that required to complete all the jobs.

*Our focus and results.* This paper studies the online machine minimization problem where the jobs have real release times, uniform processing times and equal deadlines. First, we explain why real release times and deadlines will lead to a unbounded competitive ratio for any no-waiting algorithm, and why equal deadlines, our setting, will avoid this worst case. Then, we consider two types of lookahead models and investigate how the two models improve the competitive ratios of online algorithms, respectively.

The first type (called *additive lookahead*) is the setting in which the algorithm has a constant lookahead of $L$, that is, the algorithm has the ability to know all the future information before time $t+L$ at any time $t$. We propose a $\frac{e}{\alpha(e-1)+1}$-competitive online algorithm with the lookahead $L \leq T$, where $\alpha = \frac{L}{T}$ and $T$ is the common deadline of jobs. We also present a lower bound for the problem, which shows that the algorithm is optimal for $\alpha = 0$ and $\alpha = 1$. As we can see from Fig. 1, the higher the lookahead $L$ is, the lower the competitive ratio and the lower bound will be. Particularly, the algorithm is $e$-competitive for $\alpha = 0$, which matches the prior result for the case of unit processing times and integer release times and deadlines. For $\alpha = 1$, the competitive ratio is 1, which means the algorithm outputs an optimal solution when knowing full information of the future.

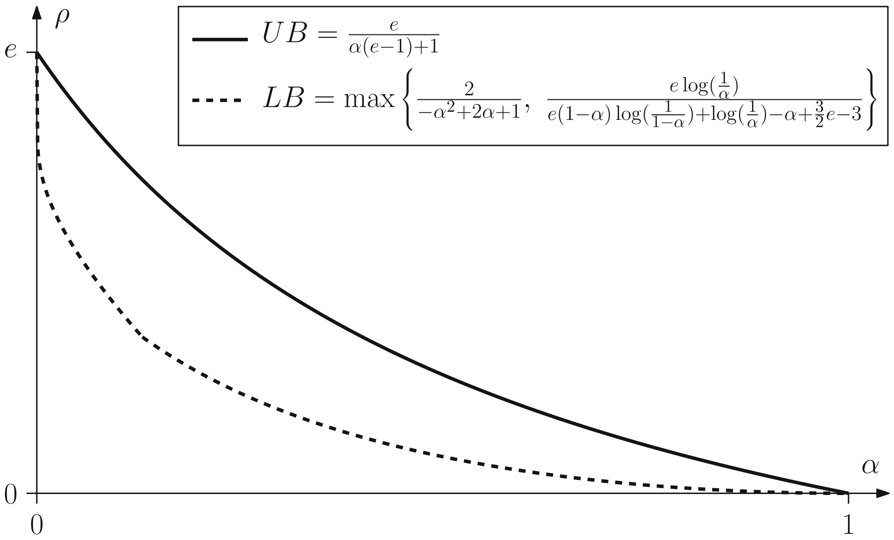


**Fig. 1** Competitive ratio and lower bound for the additive lookahead model

The second type (called *multiplicative lookahead*) models the lookahead as a function of the current time $t$, that is, at any time $t$ the algorithm has the information of all the jobs to be released before time $\beta \cdot t$, where $\beta \geq 1$. We adopt this setting because the forecasting of the future is often based on the historical data, and the more date we have the better forecasting we can do. Thus, we set the lookahead as $\beta \cdot t$, which increases as the time goes by. In this setting, we obtain an online algorithm with a competitive ratio of $\frac{\beta e}{(\beta-1)e+1}$, and provide a lower bound showing that the algorithm is optimal for $\beta = 1$ and $\beta = \infty$. The competitive ratio and the lower bound decrease in the lookahead $\beta$ (as shown in Fig. 2), and match $e$ and 1 when $\beta = 1$ and $\beta \to \infty$, respectively.

## 2 Problem definition and preliminary results

### 2.1 Problem definition

The machine minimization problem is to schedule a set of jobs on the minimum number of machines required. Denote the set of jobs by $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$, where each job $J_i$ has a release time $r_i \in \mathbb{R}$, a processing time $p_i = 1$ and a common deadline $T \in \mathbb{R}$. Note that $r_i + p_i \leq T$ for $i = 1, 2, \ldots, n$. In the online setting, each job is known to the algorithm at the release time $r_i$. The *online algorithm* decides when and to which machine each released job is assigned, and when to activate a new machine. Each job can only be processed by one machine, and each machine can only process one job at a time. The goal of a algorithm is to schedule all jobs on a minimum number of machines so that each job starts no earlier than its release time and completes by its deadline.

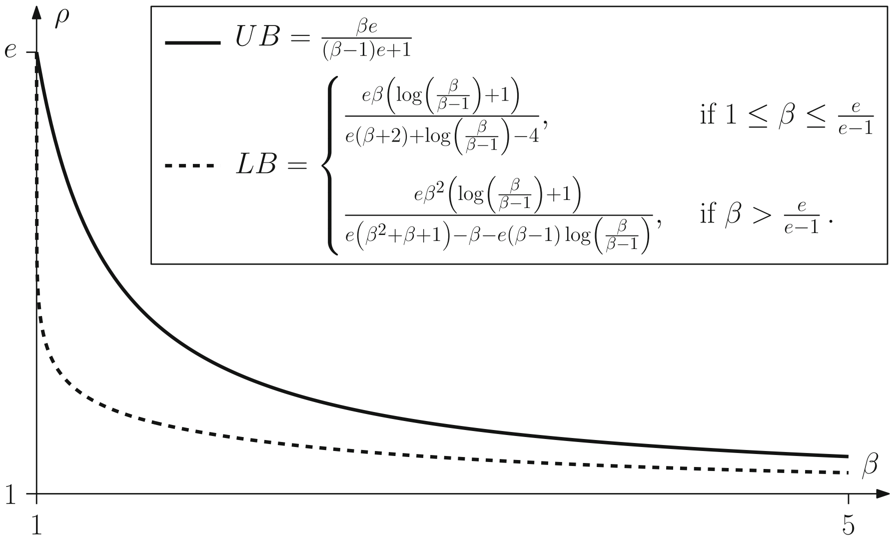


**Fig. 2** Competitive ratio and lower bound for the multiplicative lookahead model

On the contrary, the so called *offline algorithm* knows full information of the future. Thus, the offline algorithm can compute the optimal number of machines and activate such number of machines at the very beginning to ensure all jobs are processed by their deadlines. To measure the efficiency of an online algorithm, we often adopt the worst-case ratio of the number of machines activated by the online algorithm to that of the offline algorithm, that is, the *competitive ratio*. Let $opt(\mathcal{J})$ be the optimal number of machines required in order to ensure all jobs of $\mathcal{J}$ are processed by their deadline, and $C_A(\mathcal{J})$ be the number of machines that online algorithm $A$ activates for jobs of $\mathcal{J}$. An algorithm $A$ is called $\rho$-competitive (i.e., the competitive ratio is $\rho$) if there exists a constant $\delta$ such that $C_A(\mathcal{J}) \leq \rho \cdot opt(\mathcal{J}) + \delta$ for any possible set $\mathcal{J}$ of jobs.

The two types of lookahead models considered in this paper are defined in the following.

1. *Additive lookahead*: At any time $t$, the online algorithm has the ability to know all the future information before time $t + L$, where $0 \leq L \leq T$.
2. *Multiplicative lookahead*: At any time $t$, the online algorithm has the ability to know all the future information before time $\beta \cdot t$, where $\beta \geq 1$.

### 2.2 Lower bound for the case of real deadlines

**Proposition 1** *For the online machine minimization problem, where the jobs have real release times and deadlines, no no-waiting online algorithm can achieve a bounded competitive ratio. The no-waiting online algorithm refers to the algorithm who will always allocate released but unprocessed jobs to idle machines, if there is any.*

***Proof*** This proof is done by contradiction. Suppose there is a no-waiting online algorithm $A$ that has a bounded competitive ratio $\rho$. We shall derive a contradiction by providing an instance in which the online algorithm can not process all the jobs using $\rho m^*$ machines, where $m^*$ is the optimal number of machines for the offline algorithm. Hence, we can conclude that no no-waiting online algorithm for this problem can achieve a bounded competitive ratio.

Suppose the online algorithm has activated $m_1$ ($\leq \rho m^*$) machines and all the machines are idle at time $t_1$. We release $\rho m^*$ jobs who have unit processing times 1, release times $t_1$ and long deadlines $d_1$ ($>> t_1$). At the arrival of the jobs, the algorithm will probably activate some new machines and the number of activated machines is now $m_2$ ($\leq \rho m^*$). Since the algorithm will not wait, it will immediately allocate $m_2$ jobs to the $m_2$ machines. Then we release $m^*$ jobs at time $t_1 + \epsilon$ ($0 < \epsilon < 1$), who have unit processing times 1, release times $t_1 + \epsilon$ and short deadlines $t_1 + 1 + \epsilon$. Since the $m^*$ jobs must be processed at time $t_1 + \epsilon$ to be completed by the deadlines, the online algorithm must activate $m^*$ more machines to process these jobs. Hence, if $m_2 > \rho m^* - m^*$, the online algorithm will activate more than $\rho m^*$ machines to ensure all jobs are processed by their deadline, which is a contradiction. If $m_2 \leq \rho m^* - m^*$, we could apply this instance several times and eventually the number of activated machines will be more than $\rho m^*$. □

This proposition explains why real release times and deadlines will lead to a unbounded competitive ratio for all no-waiting algorithms. Obviously, the assumption of jobs with a common deadline will avoid this worst case.

### 2.3 Characterization of the optimum

Denote by $\mathcal{J}_{[t_1,t_2]}$ the set of jobs whose release times are in the interval $[t_1, t_2]$, i.e. $\mathcal{J}_{[t_1,t_2]} = \{J_i(r_i)|t_1 \leq r_i \leq t_2\}$. Likewise, define $\mathcal{J}_{[t_1,t_2)} = \{J_i(r_i)|t_1 \leq r_i < t_2\}$, $\mathcal{J}_{(t_1,t_2]} = \{J_i(r_i)|t_1 < r_i \leq t_2\}$ and $\mathcal{J}_{(t_1,t_2)} = \{J_i(r_i)|t_1 < r_i < t_2\}$. Note that the number of jobs in $\mathcal{J}_{[t_1,t_2]}$, $|\mathcal{J}_{[t_1,t_2]}|$, is also the total processing time of the jobs, since each job has a processing time of 1. The optimal number of machines is shown in the following proposition.

**Proposition 2** *For a job set $\mathcal{J}$, the optimal number of machines is*

$$opt(\mathcal{J}) = \left\lceil \max_{i=1,2,\ldots,n} \left\{ \frac{|\mathcal{J}_{[r_i,T)}|}{\lfloor T - r_i \rfloor} \right\} \right\rceil = \left\lceil \max_{k=1,2,\ldots,\lfloor T \rfloor} \left\{ \frac{|\mathcal{J}_{(T-k-1,T)}|}{k} \right\} \right\rceil$$

*Proof.* (1) The proof of $opt(\mathcal{J}) = \left\lceil \max_{i=1,2,\ldots,n} \left\{ \frac{|\mathcal{J}_{[r_i,T)}|}{\lfloor T-r_i \rfloor} \right\} \right\rceil$.

Suppose $p = \arg\max_{i=1,2,\ldots,n} \left\{ \frac{|\mathcal{J}_{[r_i,T)}|}{\lfloor T-r_i \rfloor} \right\}$. First of all, the optimal number of machines is at least $\left\lceil \frac{|\mathcal{J}_{[r_p,T)}|}{\lfloor T-r_p \rfloor} \right\rceil$, otherwise it is obviously not enough for the jobs of $\mathcal{J}_{[r_p,T)}$. Then we only have to ensure that $\left\lceil \frac{|\mathcal{J}_{[r_p,T)}|}{\lfloor T-r_p \rfloor} \right\rceil$ machines can accomplish all the jobs in $\mathcal{J}$ before the deadline $T$.

In the following, we show how to schedule all the job to $\left\lceil \frac{|\mathcal{J}_{[r_p,T)}|}{\lfloor T-r_p \rfloor} \right\rceil$ machines, so that every job starts no earlier than its release time and completes by the deadline

$T$. We divide the time line from $T - \lfloor T \rfloor$ to $T$ into $\lfloor T \rfloor$ unit length intervals. Since $p = \arg\max_{i=1,2,\ldots,n} \left\{ \frac{|\mathcal{J}_{[r_i,T)}|}{\lfloor T - r_i \rfloor} \right\}$, we know that for any starting time $t$ of an interval, the jobs released between $(t-1, T]$ need at most $\left\lceil \frac{|\mathcal{J}_{[r_p,T)}|}{\lfloor T - r_p \rfloor} \right\rceil$ machines to be completed by the deadline $T$. Thus we just allocate each job to the first available interval so that all the jobs can be processed after its release time and completed by the deadline $T$.

(2) The proof of $\left\lceil \max_{i=1,2,\ldots,n} \left\{ \frac{|\mathcal{J}_{[r_i,T)}|}{\lfloor T - r_i \rfloor} \right\} \right\rceil = \left\lceil \max_{k=1,2,\ldots,\lfloor T \rfloor} \left\{ \frac{|\mathcal{J}_{(T-k-1,T)}|}{k} \right\} \right\rceil$.

For $k = 1, 2, \ldots, \lfloor T \rfloor$, we have

$$\begin{aligned} \max_{T-k-1<r_i\le T-k} \left\{ \frac{|\mathcal{J}_{[r_i,T)}|}{\lfloor T - r_i \rfloor} \right\} &= \max_{T-k-1<r_i\le T-k} \left\{ \frac{|\mathcal{J}_{[r_i,T)}|}{k} \right\} \\ &= \frac{|\mathcal{J}_{(T-k-1,T)}|}{k}. \end{aligned}$$

Therefore, it holds that

$$\begin{aligned} \max_{i=1,2,\ldots,n} \left\{ \frac{|\mathcal{J}_{[r_i,T)}|}{\lfloor T - r_i \rfloor} \right\} &= \max_{k=1,2,\ldots,\lfloor T \rfloor} \left\{ \max_{T-k-1<r_i\le T-k} \left\{ \frac{|\mathcal{J}_{[r_i,T)}|}{\lfloor T - r_i \rfloor} \right\} \right\} \\ &= \max_{k=1,2,\ldots,\lfloor T \rfloor} \left\{ \frac{|\mathcal{J}_{(T-k-1,T)}|}{k} \right\}. \end{aligned}$$ □

Proposition 2 indicates that, for any $k = 1, 2, \ldots, \lfloor T \rfloor$, an optimal way to allocate the jobs of $\mathcal{J}_{(T-k-1,T-k]}$ is to wait until time $T - k$, and make the allocation at time $T - k$. If we also let the online algorithm make decision at time $T - k$ (for any $k = 1, 2, \ldots, \lfloor T \rfloor$), this problem is equivalent to that all the jobs have integer release times and deadlines. Since the online and offline algorithm will not make any decision in the time interval $[0, T - \lfloor T \rfloor]$, we can normalize each job $J_i$ by taking $r_i' = (r_i - T - \lfloor T \rfloor)^+$ and $T' = T - (T - \lfloor T \rfloor) = \lfloor T \rfloor$, where $r_i'$ and $T'$ are the release time and deadline of the normalized job $J_i'$. The normalized jobs have an integer deadline $T' = \lfloor T \rfloor$. Therefore, to simplify the analysis, we can assume that $T \in \mathbb{N}$.

## 3 Additive lookahead

This section considers the online machine minimization problem with the additive lookahead, that is, the algorithm, at time $t$, knows the information of jobs to be released before time $t + L$.

We define two functions:

$$f(t) = \left|\mathcal{J}_{(\lfloor t \rfloor - 1, \lfloor t \rfloor]}\right| \tag{1}$$

$$g(t) = \max_{0\le p\le t} \left\{ \frac{\int_p^t f(y)\mathrm{d}y}{T - p} \right\}, \tag{2}$$

where $0 \le t \le T$.

**Proposition 3** *Functions $f(t)$ and $g(t)$ have the following properties:*

1. *$\int_0^T f(x)\mathrm{d}x = |\mathcal{J}|$ is the total processing time of the jobs of $\mathcal{J}$;*
2. *$g(t)$ increases in $t$;*
3. *$\lceil g(t) \rceil = opt(\mathcal{J}_{[0,\lfloor t \rfloor]})$.*
4. *$\lceil g(T) \rceil = opt(\mathcal{J})$.*

***Proof*** (1) The first property is straightforward by definition.

(2) For the second property, we suppose $p = p'$ maximizes $\frac{\int_p^t f(y)\mathrm{d}y}{T-p}$ and $p = p''$ maximizes $\frac{\int_p^{t+\Delta} f(y)\mathrm{d}y}{T-p}$, where $\Delta > 0$. Thus, we have $g(t) = \frac{\int_{p'}^t f(y)\mathrm{d}y}{T-p'}$ and $g(t+\Delta) = \frac{\int_{p''}^{t+\Delta} f(y)\mathrm{d}y}{T-p''}$. Obviously, it holds that

$$g(t+\Delta) = \frac{\int_{p''}^{t+\Delta} f(y)\mathrm{d}y}{T-p''} \ge \frac{\int_{p'}^{t+\Delta} f(y)\mathrm{d}y}{T-p'} \ge g(t)\,.$$

Therefore, we know that $g(t)$ increases in $t$.

(3) We then prove the property $\lceil g(t) \rceil = opt(\mathcal{J}_{[0,\lfloor t \rfloor]})$.

Suppose $p = p'$ maximizes $\frac{\int_p^t f(y)\mathrm{d}y}{T-p}$, and thus

$$g(t) = \frac{\int_{p'}^t f(y)\mathrm{d}y}{T-p'}$$

Note that when $p'$ can take multiple values, we let $p'$ be an integer preferentially. We first prove that $p' \in \mathbb{N}^+$ by contradiction. Assuming that $p'$ is not an integer, it holds that

$$\frac{\int_{p'}^t f(y)\mathrm{d}y}{T-p'} > \frac{\int_{\lceil p' \rceil}^t f(y)\mathrm{d}y}{T-\lceil p' \rceil} = \frac{\int_{p'}^t f(y)\mathrm{d}y - (\lceil p' \rceil - p')f(p')}{(T-p') - (\lceil p' \rceil - p')}\,, \tag{3}$$

$$\frac{\int_{p'}^t f(y)\mathrm{d}y}{T-p'} > \frac{\int_{\lfloor p' \rfloor}^t f(y)\mathrm{d}y}{T-\lfloor p' \rfloor} = \frac{\int_{p'}^t f(y)\mathrm{d}y + (p' - \lfloor p' \rfloor)f(p')}{(T-p') + (p' - \lfloor p' \rfloor)}\,, \tag{4}$$

since $p = p'$ maximizes $\frac{\int_p^t f(y)\mathrm{d}y}{T-p}$ and $f(y)$ is the same value for $\lfloor p' \rfloor \le y \le \lceil p' \rceil$. Note that

According to (3) we have:

$$\frac{\int_{p'}^t f(y)\mathrm{d}y}{T-p'} > \frac{\int_{p'}^t f(y)\mathrm{d}y - (\lceil p' \rceil - p')f(p')}{(T-p') - (\lceil p' \rceil - p')}$$

$$\Rightarrow \quad -(\lceil p' \rceil - p')\int_{p'}^t f(y)\mathrm{d}y > -(T-p')(\lceil p' \rceil - p')f(p')$$

$$\Rightarrow \quad \int_{p'}^{t} f(y)\mathrm{d}y < (T - p')f(p'). \tag{5}$$

According to (4) we have:

$$\begin{aligned}
&\frac{\int_{p'}^{t} f(y)\mathrm{d}y}{T - p'} > \frac{\int_{p'}^{t} f(y)\mathrm{d}y + (p' - \lfloor p' \rfloor) f(p')}{(T - p') + (p' - \lfloor p' \rfloor)} \\
\Rightarrow \quad & (p' - \lfloor p' \rfloor) \int_{p'}^{t} f(y)\mathrm{d}y > (T - p')(p' - \lfloor p' \rfloor) f(p') \\
\Rightarrow \quad & \int_{p'}^{t} f(y)\mathrm{d}y > (T - p') f(p').
\end{aligned} \tag{6}$$

The two inequalities (5) and (6) show a contradiction. Hence, we obtain that $p' \in \mathbb{N}^+$. Therefore, it holds that

$$\begin{aligned}
g(t) &= \max_{0 \le p \le t} \left\{ \frac{\int_p^t f(y)\mathrm{d}y}{T - p} \right\} \\
&= \max_{p = 0, 1, \ldots, \lfloor t \rfloor} \left\{ \frac{\int_p^t \left| \mathcal{J}_{(\lfloor y \rfloor - 1, \lfloor y \rfloor]} \right| \mathrm{d}y}{T - p} \right\} \\
&= \max_{p = 0, 1, \ldots, \lfloor t \rfloor} \left\{ \frac{\left| \mathcal{J}_{(p-1, \lfloor t \rfloor]} \right|}{T - p} \right\} \\
&= \max_{k = T - \lfloor t \rfloor, T - \lfloor t \rfloor + 1, \ldots, T} \left\{ \frac{\left| \mathcal{J}_{(T-k-1, \lfloor t \rfloor]} \right|}{k} \right\}.
\end{aligned}$$

Since $opt(\mathcal{J}) = \left\lceil \max_{k=1,2,\ldots,\lfloor T \rfloor} \left\{ \frac{|\mathcal{J}_{(T-k-1,T)}|}{k} \right\} \right\rceil$ (Proposition 2), we have

$$\begin{aligned}
opt(\mathcal{J}_{[0, \lfloor t \rfloor]}) &= \left\lceil \max_{k=1,2,\ldots,\lfloor T \rfloor} \left\{ \frac{\left| \mathcal{J}_{(T-k-1, \lfloor t \rfloor]} \right|}{k} \right\} \right\rceil \\
&= \left\lceil \max_{k = T - \lfloor t \rfloor, T - \lfloor t \rfloor + 1, \ldots, \lfloor T \rfloor} \left\{ \frac{\left| \mathcal{J}_{(T-k-1, \lfloor t \rfloor]} \right|}{k} \right\} \right\rceil.
\end{aligned}$$

Hence, it holds that $\lceil g(t) \rceil = opt(\mathcal{J}_{[0, \lfloor t \rfloor]})$, as $T \in \mathbb{N}^+$.

(4) The fourth property can be deduced from the third property by taking $t$ as $T$. □

In the following, we give the Algorithm A.

**Algorithm A:** At any time $t$, algorithm A always keeps a function of $\lceil A(t) \rceil$ machines, and assigns any unprocessed job to some machine whenever a machine becomes idle. $A(t)$ is defined as:

$$A(t) = \begin{cases} \rho \cdot g(t+L), & \text{if } 0 \le t \le T-L \\ \rho \cdot \lceil g(T) \rceil, & \text{if } T-L < t \le T\,, \end{cases}$$

where $\rho = \frac{e}{\alpha(e-1)+1}$ and $\alpha = \frac{L}{T}$.

**Theorem 4** *Algorithm A is* $\frac{e}{\alpha(e-1)+1}$*-competitive for the online machine minimization problem with the additive lookahead, where* $\alpha = \frac{L}{T}$.

***Proof*** Since algorithm A activates $\lceil A(T) \rceil$ machines at last (Proposition 3, property 2) and

$$\lceil A(T) \rceil = \lceil \rho \cdot \lceil g(T) \rceil \rceil \le \rho \cdot \lceil g(T) \rceil + 1 = \rho \cdot opt(\mathcal{J}) + 1\,,$$

we only need to prove that $\lceil A(t) \rceil$ machines can process all the jobs by the deadline. **Case 1:** There is no idle time during the whole process, i.e. every machine is busy from its activation time to the deadline $T$. We will show that $\int_0^T A(t)\mathrm{d}t \ge \int_0^T f(t)\mathrm{d}t$, which indicates that all jobs of $\mathcal{J}$ can be processed by the deadline.

By definition, we have

$$\begin{aligned} \int_0^T A(t)\mathrm{d}t &= \rho \int_0^{T-L} g(t+L)\mathrm{d}t + \rho \int_{T-L}^{T} \lceil g(T) \rceil \mathrm{d}t \\ &= \rho \int_0^{T-L} \max_{0 \le p \le t+L} \left\{ \frac{\int_p^{t+L} f(y)\mathrm{d}y}{T-p} \right\} \mathrm{d}t + \rho \cdot L \cdot opt(\mathcal{J})\,. \end{aligned}$$

Suppose $p = p(t)$ that maximizes $\frac{\int_p^{t+L} f(y)\mathrm{d}y}{T-p}$. Thus

$$\int_0^T A(t)\mathrm{d}t = \rho \int_0^{T-L} \frac{\int_{p(t)}^{t+L} f(y)\mathrm{d}y}{T-p(t)} \mathrm{d}t + \rho \cdot L \cdot opt(\mathcal{J})\,. \tag{7}$$

In order to derive $\int_0^T f(t)\mathrm{d}t$ from the double integral $\int_0^{T-L} \frac{\int_{p(t)}^{t+L} f(y)\mathrm{d}y}{T-p(t)}\mathrm{d}t$, we reduce the integration domain as shown in Fig. 3. Thus, we have

$$\begin{aligned} \int_0^{T-L} \frac{\int_{p(t)}^{t+L} f(y)\mathrm{d}y}{T-p(t)} \mathrm{d}t &\ge \int_0^{T-L} \frac{\int_{p(t)}^{\frac{T}{T-L}t} f(y)\mathrm{d}y}{T-p(t)} \mathrm{d}t \\ &= \int_0^T f(y) \int_{\frac{T-L}{T}y}^{p^{-1}(y)} \frac{1}{T-p(t)} \mathrm{d}t\mathrm{d}y\,. \end{aligned}$$

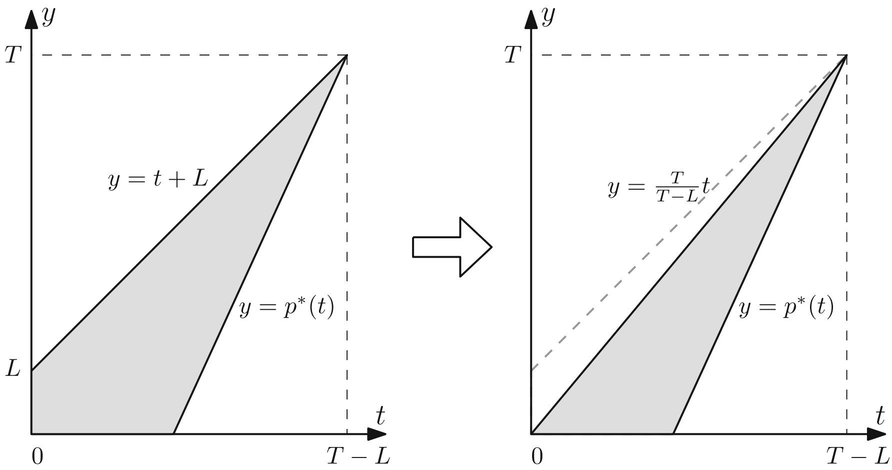


**Fig. 3** Reduction of the integration domain

Since $p = p(t)$ maximizes $\frac{\int_p^{t+L} f(y)\mathrm{d}y}{T-p}$, we can obtain the following inequality by taking $p = T - e(T - \frac{T}{T-L}t)$:

$$\int_{\frac{T-L}{T}y}^{p^{-1}(y)} \frac{1}{T - p(t)}\mathrm{d}t \geq \int_{\frac{T-L}{T}y}^{\frac{T-L}{eT}y+(1-\frac{1}{e})(T-L)} \frac{1}{e(T - \frac{T}{T-L}t)}\mathrm{d}t = \frac{T-L}{eT} .$$

Thus

$$\int_0^{T-L} \frac{\int_{p(t)}^{t+L} f(y)\mathrm{d}y}{T - p(t)}\mathrm{d}t \geq \frac{T-L}{eT} \int_0^T f(y)\mathrm{d}y .$$

Since $opt(\mathcal{J}) \geq \frac{1}{T}\int_0^T f(y)\mathrm{d}y$, we can deduce from (7) that

$$\begin{aligned}\int_0^T A(t)\mathrm{d}t &= \rho \int_0^{T-L} \frac{\int_{p(t)}^{t+L} f(y)\mathrm{d}y}{T - p(t)}\mathrm{d}t + \rho \cdot L \cdot opt(\mathcal{J}) \\ &\geq \rho \frac{T-L}{eT} \int_0^T f(y) + \rho \frac{L}{T} \int_0^T f(y)\mathrm{d}y \\ &= \int_0^T f(y)\mathrm{d}y ,\end{aligned}$$

where $\rho = \frac{e}{\alpha(e-1)+1}$ and $\alpha = \frac{L}{T}$. Therefore, we have $\int_0^T A(t)\mathrm{d}t \geq \int_0^T f(y)\mathrm{d}y$.

**Case 2:** There is some idle time during the whole process. Suppose the last idle time finishes at $t_f$, which means that during the interval $[t_f, T]$ all machines are busy. Obviously, at time $t_f$ all the jobs arrived before $t_f$ are completed, otherwise the last idle time would not exist. Therefore, the time $t_f$ can be seen as a new beginning without any unfinished jobs, and the case of time period $[t_f, T]$ can be done by the approach of Case 1. □

## 4 Multiplicative lookahead

In this section, we consider the case of the multiplicative lookahead, that is, the online algorithm knows the future information before time $\beta t$ at any time $t$, where $\beta \geq 1$.
**Algorithm B:** At any time $t$, algorithm B always keeps a function of $\lceil B(t) \rceil$ machines, and assigns any unprocessed job to some machine whenever a machine becomes idle. $B(t)$ is defined as:

$$B(t) = \begin{cases} \rho \cdot g(\beta t), & \text{if } 0 \leq t \leq \frac{T}{\beta} \\ \rho \cdot \lceil g(T) \rceil, & \text{if } \frac{T}{\beta} < t \leq T \end{cases},$$

where $\rho = \frac{\beta e}{(\beta-1)e+1}$ and function $g(x)$ is defined by (1) and (2).

**Theorem 5** *Algorithm B is $\frac{\beta e}{(\beta-1)e+1}$-competitive for the online machine minimization problem with the multiplicative lookahead.*

***Proof*** Because algorithm $B$ will always keep no more than $\lceil \rho \cdot \lceil g(T) \rceil \rceil$ machines $\left(\rho = \frac{\beta e}{1+(\beta-1)e}\right)$, the algorithm certainly guarantees a competitive ratio of $\rho$, if all the jobs can be completed in these machines. Therefore, we just need to prove that these machines can complete all the jobs by their deadlines.
**Case 1:** There is no idle time during the whole process, i.e., every machine is busy from its activation time to the deadline $T$. We will show that $\int_0^T B(t)\mathrm{d}t \geq \int_0^T f(t)\mathrm{d}t$, which indicates that all the jobs of $\mathcal{J}$ can be processed by the deadlines.

Similar to the proof of Theorem 4, we have

$$\begin{aligned}
\int_0^T B(t)\mathrm{d}t &= \rho \int_0^{\frac{T}{\beta}} g(\beta t)\mathrm{d}t + \rho \int_{\frac{T}{\beta}}^T \lceil g(T) \rceil \mathrm{d}t \\
&= \rho \int_0^{\frac{T}{\beta}} \frac{\int_{p(t)}^{\beta t} f(y)\mathrm{d}y}{T - p(t)} \mathrm{d}t + \rho \int_{\frac{T}{\beta}}^T \lceil g(T) \rceil \mathrm{d}t \\
&\geq \rho \int_0^{\frac{T}{\beta}} \frac{\int_{T-e(T-\beta t)}^{\beta t} f(y)\mathrm{d}y}{e(T - \beta t)} \mathrm{d}t + \rho \int_{\frac{T}{\beta}}^T \lceil g(T) \rceil \mathrm{d}t \\
&= \frac{\rho}{e} \int_0^T f(y) \int_{\frac{x}{\beta}}^{\frac{T}{\beta} - \frac{T-x}{e\beta}} \frac{1}{T - \beta t} \mathrm{d}t\mathrm{d}y + \rho \left(T - \frac{T}{\beta}\right) opt(\mathcal{J}) \\
&= \frac{\rho}{e\beta} \int_0^T f(y)\mathrm{d}y + \rho \left(T - \frac{T}{\beta}\right) opt(\mathcal{J}) \\
&\geq \frac{\rho}{e\beta} \int_0^T f(y)\mathrm{d}y + \rho \left(T - \frac{T}{\beta}\right) \frac{\int_0^T f(y)\mathrm{d}y}{T}
\end{aligned}$$

$$= \int_0^T f(y)\mathrm{d}y$$

Therefore, we obtain the conclusion that $\int_0^T B(t)\mathrm{d}t \geq \int_0^T f(t)\mathrm{d}t$.
**Case 2:** There is some idle time during the whole process. Suppose the last idle time finishes at $t_f$, which means that during the interval $[t_f, T]$ all machines are busy. Obviously, at time $t_f$ all the jobs arrived before $t_f$ are completed, otherwise the last idle time would not exist. Therefore, the time $t_f$ can be seen as a new beginning without any unfinished jobs, and the case of time period $[t_f, T]$ can be done by the approach of Case 1. □

## 5 Lower bounds with lookahead

In this section, we investigate the lower bounds of the problem, i.e. the smallest possible competitive ratio an online algorithm can obtain.

### 5.1 Lower bound for the additive lookahead model

**Theorem 6** *Any $\rho$-competitive algorithm for the online machine minimization problem with the additive lookahead $\alpha = \frac{L}{T}$ has*

$$\rho \geq \max\left\{\frac{2}{-\alpha^2 + 2\alpha + 1}, \frac{e\log(\frac{1}{\alpha})}{e(1-\alpha)\log(\frac{1}{1-\alpha}) + \log(\frac{1}{\alpha}) - \alpha + \frac{3}{2}e - 3}\right\}$$

*Particularly, when $\alpha \to 0$ (without lookahead), the lower bound is e, and when $\alpha \to 1$ (with full information) the lower bound is* 1.

***Proof*** Suppose $H$ is a $\rho$-competitive online algorithm. At time $t$, the number of online machines is denoted by $H(t)$. To guarantee the algorithm is $\rho$-competitive, it satisfies that

$$H(t) \leq \begin{cases} \rho \cdot opt(\mathcal{J}_{[0,t+L]}), & \text{if } t \in [0, T-L] \\ \rho \cdot opt(\mathcal{J}), & \text{if } t \in [T-L, T] \end{cases}$$

at any time $t \in [0, T]$.

We define a set $\mathcal{J}$ of jobs, which satisfies

$$f(t) = \begin{cases} \left\lfloor \frac{T^2}{T - \lfloor t \rfloor} \right\rfloor, & \text{if } 0 \leq \lfloor t \rfloor \leq T - \delta \\ \left\lfloor \frac{T^2}{\delta} \right\rfloor, & \text{if } T - \delta < \lfloor t \rfloor \leq T\,, \end{cases} \tag{8}$$

where $f(t) = \left|\mathcal{J}_{(\lfloor t \rfloor - 1, \lfloor t \rfloor]}\right|$ and $\delta$ can take any value in $[0, T]$. Note that each job in $\mathcal{J}$ is released at integer time point, i.e., $f(k)$ jobs are released at time $k = 0, 1, \ldots, T$.

Recall that $g(t) = \max_{0\le p\le t}\left\{\frac{\int_p^t f(y)\mathrm{d}y}{T-p}\right\}$ and $\lceil g(t)\rceil = opt(\mathcal{J}_{[0,\lfloor t\rfloor]})$. Since each job is released at integer time point, we have

$$opt(\mathcal{J}_{[0,t]}) = opt(\mathcal{J}_{[0,\lfloor t\rfloor]}) = \lceil g(t)\rceil \le g(t) + 1\,.$$

Therefore, it holds that

$$\frac{H(t)}{\rho} \le \begin{cases} g(t+L)+1\,, & t\in[0,T-L] \\ g(T)+1\,, & t\in[T-L,T] \end{cases}$$

1) We first provide a general lower bound for $0\le L\le T$.
Taking $\delta = T$ (in (8)), it holds that

$$f(t) = T\,.$$

We characterize the upper bound for $\frac{h(t)}{\rho}$:

$$\begin{aligned}
\frac{H(t)}{\rho} - 1 &\le \begin{cases} g(t+L), & \text{if } t\in[0,T-L] \\ g(T), & \text{if } t\in[T-L,T] \end{cases} \\
&\le \begin{cases} \max_{0\le p\le t+L}\left\{\frac{\int_p^{t+L} T\mathrm{d}y}{T-p}\right\}, & \text{if } t\in[0,T-L] \\ \max_{0\le p\le T}\left\{\frac{\int_p^{T} T\mathrm{d}y}{T-p}\right\}, & \text{if } t\in[T-L,T] \end{cases} \\
&= \begin{cases} t+L, & \text{if } t\in[0,T-L] \\ T, & \text{if } t\in[T-L,T] \end{cases}
\end{aligned}$$

Hence, it follows that

$$\begin{aligned}
\frac{1}{\rho}\int_0^T H(t)\mathrm{d}t &\le \int_0^{T-L}(t+L+1)\mathrm{d}t + \int_{T-L}^{T}(T+1)\mathrm{d}t \\
&= -\frac{L^2}{2} + LT + \frac{1}{2}T(T+2)\,.
\end{aligned} \tag{9}$$

Because the total jobs that online machines process must be no less than the total released jobs, it holds that

$$\int_0^T H(t)\mathrm{d}t \ge \int_0^T f(t)\mathrm{d}t = T^2\,. \tag{10}$$

According to (9) and (10), it follows that

$$\rho \geq \frac{T^2}{-\frac{L^2}{2} + LT + \frac{1}{2}T(T+2)} = \frac{2}{-\alpha^2 + 2\alpha + 1 + \frac{2}{T}},$$

where $\alpha = \frac{L}{T}$. Taking $T \to \infty$, we have

$$\rho \geq \frac{2}{-\alpha^2 + 2\alpha + 1}.$$

2) We then give a better lower bound for $L \leq \frac{T}{e^2}$.
Taking $\delta = eL$ (in (8)), it follows that

$$f(t) = \begin{cases} \left\lfloor \frac{T^2}{T - \lfloor t \rfloor} \right\rfloor, & \text{if } 0 \leq \lfloor t \rfloor \leq T - eL \\ \left\lfloor \frac{T^2}{eL} \right\rfloor, & \text{if } T - eL < \lfloor t \rfloor \leq T. \end{cases}$$

We define a upper bound and a lower bound of $f(t)$, respectively:

$$f(t) \leq \overline{f}(t) = \begin{cases} \frac{T^2}{T - \lfloor t \rfloor}, & \text{if } 0 \leq t \leq T - eL \\ \frac{T^2}{eL}, & \text{if } T - eL < t \leq T. \end{cases}$$

$$f(t) \geq \underline{f}(t) = \begin{cases} \frac{T^2}{T-(t-1)} - 1, & \text{if } 0 \leq t \leq T - eL + 1 \\ \frac{T^2}{eL} - 1, & \text{if } T - eL + 1 < t \leq T. \end{cases}$$

We characterize the upper bound for $\frac{h(t)}{\rho}$.

(a) For $t \leq T - L - eL$, we have

$$\begin{aligned} \frac{H(t)}{\rho} - 1 &\leq g(t+L) \\ &\leq \max_{0 \leq p \leq t+L} \left\{ \frac{\int_p^{t+L} \overline{f}(y)\mathrm{d}y}{T-p} \right\} \\ &= \max_{0 \leq p \leq t+L} \left\{ \frac{\int_p^{t+L} \frac{T^2}{T-y}\mathrm{d}y}{T-p} \right\} \\ &= \max_{0 \leq p \leq t+L} \left\{ \frac{T^2\left(\log\left(1-\frac{p}{T}\right) - \log\left(1 - \frac{L+t}{T}\right)\right)}{T-p} \right\} \\ &= \begin{cases} T\log\left(\frac{T}{T-L-t}\right), & \text{if } 0 \leq t \leq T - L - \frac{T}{e} \\ \frac{T^2}{e(T-L-t)}, & \text{if } T - L - \frac{T}{e} < t \leq T - L - eL, \end{cases} \end{aligned}$$

where the last equation is because $p^* = 0$ for $0 \le t \le T - L - \frac{T}{e}$ and $p^* = e(L + t - T) + T$ for $T - L - \frac{T}{e} < t \le T - L - eL$.

(b) For $T - L - eL < t \le T - L$, we have

$$
\begin{aligned}
\frac{H(t)}{\rho} - 1 &\le g(t+L) \\
&\le \max_{0 \le p \le t+L} \left\{ \frac{\int_p^{t+L} \overline{f}(y)\mathrm{d}y}{T-p} \right\} \\
&= \begin{cases} \max_{0 \le p \le T-eL} \left\{ \frac{\int_p^{T-eL} \frac{T^2}{T-y}\mathrm{d}y + \int_{T-eL}^{t+L} \frac{T^2}{eL}\mathrm{d}y}{T-p} \right\}, & \text{if } 0 \le p \le T - eL \\ \max_{T-eL < p \le t+L} \left\{ \frac{\int_p^{t+L} \frac{T^2}{eL}\mathrm{d}y}{T-p} \right\}, & \text{if } T - eL < p \le t + L \end{cases} \\
&= \frac{T^2(eL + L + t - T)}{e^2L^2},
\end{aligned}
$$

where the last equation is because $p^* = T - eL$.

(c) For $T - L < t \le T$, we have

$$
\frac{H(t)}{\rho} - 1 \le g(T) = \frac{T^2}{eL}.
$$

Therefore, we obtain that

$$
\frac{H(t)}{\rho} \le \begin{cases} T\log\left(\frac{T}{T-L-t}\right) + 1, & \text{if } 0 \le t \le T - L - \frac{T}{e} \\ \frac{T^2}{e(T-L-t)} + 1, & \text{if } T - L - \frac{T}{e} < t \le T - L - eL \\ \frac{T^2(eL+L+t-T)}{e^2L^2} + 1, & \text{if } T - L - eL < t \le T - L \\ \frac{T^2}{eL} + 1, & \text{if } T - L < t \le T. \end{cases}
$$

Hence, it follows that

$$
\begin{aligned}
\frac{1}{\rho}\int_0^T H(t)\mathrm{d}t \le & \int_0^{T-L-\frac{T}{e}} \left(T\log\left(\frac{T}{T-L-t}\right) + 1\right)\mathrm{d}t \\
& + \int_{T-L-\frac{T}{e}}^{T-L-eL} \left(\frac{T^2}{e(T-L-t)} + 1\right)\mathrm{d}t \\
& + \int_{T-L-eL}^{T-L} \left(\frac{T^2(eL+L+t-T)}{e^2L^2} + 1\right)\mathrm{d}t \\
& + \int_{T-L}^{T} \left(\frac{T^2}{eL} + 1\right)\mathrm{d}t \\
= & \; T(T-L)\log\left(\frac{T}{T-L}\right) + \frac{T^2}{e}\log\left(\frac{T}{L}\right)
\end{aligned}
$$

$$- T(L-1) + 3T^2\left(\frac{1}{2} - \frac{1}{e}\right) \tag{11}$$

Because the total jobs that online machines process must be no less than the total released jobs, it holds that

$$\begin{aligned}\int_0^T H(t)\mathrm{d}t &\geq \int_0^T f(t)\mathrm{d}t \\ &\geq \int_0^T \underline{f}(t)\mathrm{d}t \\ &= -\frac{T^2}{eL} + T^2\log\left(\frac{T+1}{L}\right) - T \end{aligned} \tag{12}$$

According to (11) and (12), we have

$$\begin{aligned}\rho &\geq \frac{-\frac{T^2}{eL} + T^2\log\left(\frac{T+1}{L}\right) - T}{T(T-L)\log\left(\frac{T}{T-L}\right) + \frac{T^2}{e}\log\left(\frac{T}{L}\right) - T(L-1) + 3T^2\left(\frac{1}{2}-\frac{1}{e}\right)} \\ &= \frac{-\frac{1}{e\alpha} + T\log\left(\frac{T+1}{\alpha T}\right) - 1}{T(1-\alpha)\log\left(\frac{1}{1-\alpha}\right) + \frac{T}{e}\log\left(\frac{1}{\alpha}\right) - (\alpha T - 1) + 3T\left(\frac{1}{2}-\frac{1}{e}\right)}.\end{aligned}$$

Taking $T \to \infty$, it follows that

$$\rho \geq \frac{e\log(\frac{1}{\alpha})}{e(1-\alpha)\log(\frac{1}{1-\alpha}) + \log(\frac{1}{\alpha}) - \alpha + \frac{3}{2}e - 3}.$$

Therefore, we conclusion that

$$\rho \geq \max\left\{\frac{2}{-\alpha^2+2\alpha+1}, \frac{e\log(\frac{1}{\alpha})}{e(1-\alpha)\log(\frac{1}{1-\alpha}) + \log(\frac{1}{\alpha}) - \alpha + \frac{3}{2}e - 3}\right\},$$

which proves the theorem. □

### 5.2 Lower bound for the multiplicative lookahead model

**Theorem 7** *Any $\rho$-competitive algorithm for the online machine minimization problem with the additive lookahead $\beta$ has*

$$\rho \geq \begin{cases} \dfrac{e\beta\left(\log\left(\frac{\beta}{\beta-1}\right)+1\right)}{e(\beta+2)+\log\left(\frac{\beta}{\beta-1}\right)-4}, & \text{if } 1 \leq \beta \leq \frac{e}{e-1} \\ \dfrac{e\beta^2\left(\log\left(\frac{\beta}{\beta-1}\right)+1\right)}{e\left(\beta^2+\beta+1\right)-\beta-e(\beta-1)\log\left(\frac{\beta}{\beta-1}\right)}, & \text{if } \beta > \frac{e}{e-1}. \end{cases}$$

*Particularly, when $\beta \to 1$ (without lookahead), the lower bound is $e$, and when $\beta \to +\infty$ (with full information) the lower bound is* 1.

***Proof*** Suppose $H$ is a $\rho$-competitive online algorithm. At time $t$, the number of online machines is denoted by $H(t)$. To guarantee the algorithm is $\rho$-competitive, it satisfies that

$$H(t) \leq \begin{cases} \rho \cdot opt(\mathcal{J}_{[0,\beta t]}), & \text{if } t \in [0, \frac{T}{\beta}] \\ \rho \cdot opt(\mathcal{J}), & \text{if } t \in [\frac{T}{\beta}, T] \end{cases} \tag{13}$$

at any time $t \in [0, T]$.

We define a set $\mathcal{J}$ of jobs, which satisfies

$$f(t) = \begin{cases} \left\lfloor \frac{T^2}{T-\lfloor t \rfloor} \right\rfloor, & \text{if } 0 \leq \lfloor t \rfloor \leq \frac{T}{\beta} \\ \left\lfloor \frac{\beta T}{\beta - 1} \right\rfloor, & \text{if } \frac{T}{\beta} < \lfloor t \rfloor \leq T\,, \end{cases} \tag{14}$$

where $f(t) = \left|\mathcal{J}_{(\lfloor t \rfloor - 1, \lfloor t \rfloor]}\right|$. Notice that each job is released at integer time point, i.e., $f(k)$ jobs are released at time $k = 0, 1, \ldots, T$.

We define a upper bound and a lower bound of $f(t)$, respectively:

$$f(t) \leq \overline{f}(t) = \begin{cases} \frac{T^2}{T-t}, & \text{if } 0 \leq t \leq \frac{T}{\beta} \\ \frac{\beta T}{\beta-1}, & \text{if } \frac{T}{\beta} < t \leq T\,. \end{cases}$$

$$f(t) \geq \underline{f}(t) = \begin{cases} \frac{T^2}{T-(t-1)} - 1, & \text{if } 0 \leq t \leq \frac{T}{\beta} + 1 \\ \frac{\beta T}{\beta-1} - 1, & \text{if } \frac{T}{\beta} + 1 < t \leq T\,. \end{cases}$$

Recall that $g(t) = \max_{0 \leq p \leq t} \left\{ \frac{\int_p^t f(y)\mathrm{d}y}{T-p} \right\}$ and $\lceil g(t) \rceil = opt(\mathcal{J}_{[0,\lfloor t \rfloor]})$. Since each job is released at integer time point, we have

$$opt(\mathcal{J}_{[0,t]}) = opt(\mathcal{J}_{[0,\lfloor t \rfloor]}) = \lceil g(t) \rceil \leq g(t) + 1\,.$$

Therefore, it holds that

$$\frac{H(t)}{\rho} \leq \begin{cases} g(\beta t) + 1\,, & t \in [0, \dfrac{T}{\beta}] \\ g(T) + 1\,, & t \in [\dfrac{T}{\beta}, T] \end{cases} \tag{15}$$

We then characterize the upper bound for $\frac{h(t)}{\rho}$.

1) We first consider the case $1 \leq \beta \leq \frac{e}{e-1}$.

(a) For $t \leq \frac{T}{\beta^2}$, we have

$$\frac{H(t)}{\rho} - 1 \leq g(\beta t)$$

$$
\begin{aligned}
&\le \max_{0\le p\le \beta t}\left\{\frac{\int_p^{\beta t}\overline{f}(y)\mathrm{d}y}{T-p}\right\}\\
&= \max_{0\le p\le \beta t}\left\{\frac{\int_p^{\beta t}\frac{T^2}{T-y}\mathrm{d}y}{T-p}\right\}\\
&= \max_{0\le p\le \beta t}\left\{\frac{T^2\log\left(\frac{T-p}{T-\beta t}\right)}{T-p}\right\}\\
&= \begin{cases} T\log\left(\frac{T}{T-\beta t}\right), & \text{if } 0\le t\le \frac{(e-1)T}{e\beta}\\ \frac{T^2}{e(T-\beta t)}, & \text{if } \frac{(e-1)T}{e\beta} < t \le \frac{T}{\beta^2}\end{cases}
\end{aligned}
$$

(b) For $\frac{T}{\beta^2} < t \le \frac{T}{\beta}$, we have

$$
\begin{aligned}
\frac{H(t)}{\rho} - 1 &\le g(\beta t)\\
&\le \max_{0\le p\le \beta t}\left\{\frac{\int_p^{\beta t}\overline{f}(y)\mathrm{d}y}{T-p}\right\}\\
&= \begin{cases} \max_{0\le p\le \frac{T}{\beta}}\left\{\frac{\int_p^{T/\beta}\frac{T^2}{T-y}\mathrm{d}y+\int_{T/\beta}^{\beta t}\frac{\beta T}{\beta-1}\mathrm{d}y}{T-p}\right\}, & \text{if } 0\le p\le \frac{T}{\beta}\\ \max_{\frac{T}{\beta}<p\le \beta t}\left\{\frac{\int_p^{\beta t}\frac{\beta T}{\beta-1}\mathrm{d}y}{T-p}\right\}, & \text{if } \frac{T}{\beta} < p \le \beta t\end{cases}\\
&= \frac{\beta T e^{\frac{\beta(\beta t-T)}{(\beta-1)T}}}{\beta-1}.
\end{aligned}
$$

(c) For $\frac{T}{\beta} < t \le T$, we have

$$
\begin{aligned}
\frac{H(t)}{\rho} - 1 &\le g(T)\\
&\le \max_{0\le p\le T}\left\{\frac{\int_p^{T}\overline{f}(y)\mathrm{d}y}{T-p}\right\}\\
&= \max_{0\le p\le T}\left\{\frac{\int_p^{T}\frac{\beta T}{\beta-1}\mathrm{d}y}{T-p}\right\}\\
&= \frac{\beta T}{\beta-1}.
\end{aligned}
$$

Therefore, we obtain that

$$\frac{H(t)}{\rho} \leq \begin{cases} T\log\left(\frac{T}{T-\beta t}\right)+1, & \text{if } 0 \leq t \leq \frac{(e-1)T}{e\beta} \\ \frac{T^2}{e(T-\beta t)}+1, & \text{if } \frac{(e-1)T}{e\beta} < t \leq \frac{T}{\beta^2} \\ \frac{\beta T e^{\frac{\beta(\beta t-T)}{(\beta-1)T}}}{\beta-1}+1, & \text{if } \frac{T}{\beta^2} < t \leq \frac{T}{\beta} \\ \frac{\beta T}{\beta-1}+1, & \text{if } \frac{T}{\beta} < t \leq T\,. \end{cases}$$

Hence, it follows that

$$\begin{aligned} \frac{1}{\rho}\int_0^T H(t)\mathrm{d}t &\leq \int_0^{\frac{(e-1)T}{e\beta}} (T\log\left(\frac{T}{T-\beta t}\right)+1)\mathrm{d}t \\ &+ \int_{\frac{(e-1)T}{e\beta}}^{\frac{T}{\beta^2}} (\frac{T^2}{e(T-\beta t)}+1)\mathrm{d}t \\ &+ \int_{\frac{T}{\beta^2}}^{\frac{T}{\beta}} (\frac{\beta T e^{\frac{\beta(\beta t-T)}{(\beta-1)T}}}{\beta-1}+1)\mathrm{d}t \\ &+ \int_{\frac{T}{\beta}}^{T} (\frac{\beta T}{\beta-1}+1)\mathrm{d}t \\ &= \frac{T\left(e\beta+(e(\beta+2)-4)T-T\log\left(\frac{\beta-1}{\beta}\right)\right)}{e\beta} \end{aligned} \tag{16}$$

Because the total jobs that online machines process must be no less than the total released jobs, that is,

$$\begin{aligned} \int_0^T H(t)\mathrm{d}t &\geq \int_0^T f(t)\mathrm{d}t \\ &\geq \int_0^T \underline{f}(t)\mathrm{d}t \\ &= T\left(-\frac{\beta}{\beta-1}+T\log\left(\frac{\beta(T+1)}{(\beta-1)T}\right)+T-1\right) \end{aligned} \tag{17}$$

According to (16) and (17), we have

$$\begin{aligned} \rho &\geq \frac{T\left(-\frac{\beta}{\beta-1}+T\log\left(\frac{\beta(T+1)}{(\beta-1)T}\right)+T-1\right)}{\frac{T\left(e\beta+(e(\beta+2)-4)T-T\log\left(\frac{\beta-1}{\beta}\right)\right)}{e\beta}} \\ &= \frac{e\beta\left(-\frac{\beta}{\beta-1}+T\log\left(\frac{\beta(T+1)}{(\beta-1)T}\right)+T-1\right)}{e\beta+(e(\beta+2)-4)T-T\log\left(\frac{\beta-1}{\beta}\right)}\,. \end{aligned}$$

Taking $T \to \infty$, it follows that

$$\rho \geq \frac{e\beta\left(\log\left(\frac{\beta}{\beta-1}\right)+1\right)}{e(\beta+2)+\log\left(\frac{\beta}{\beta-1}\right)-4}.$$

2) We then consider the case $\beta > \frac{e}{e-1}$.
Similar as above, we obtain that

$$\frac{H(t)}{\rho} \leq \begin{cases} T\log\left(\frac{T}{T-\beta t}\right)+1, & \text{if } 0 \leq t \leq \frac{T}{\beta^2} \\ \frac{\beta T e^{\frac{\beta(\beta t-T)}{(\beta-1)T}}}{\beta-1}+1, & \text{if } \frac{T}{\beta^2} < t \leq \frac{T}{\beta} \\ \frac{\beta T}{\beta-1}+1, & \text{if } \frac{T}{\beta} < t \leq T. \end{cases}$$

Thus, it follows that

$$\begin{aligned} \frac{1}{\rho}\int_0^T H(t)\mathrm{d}t \leq & \int_0^{\frac{T}{\beta^2}} (T\log\left(\frac{T}{T-\beta t}\right)+1)\mathrm{d}t \\ & + \int_{\frac{T}{\beta^2}}^{\frac{T}{\beta}} (\frac{\beta T e^{\frac{\beta(\beta t-T)}{(\beta-1)T}}}{\beta-1}+1)\mathrm{d}t \\ & + \int_{\frac{T}{\beta}}^{T} (\frac{\beta T}{\beta-1}+1)\mathrm{d}t \\ = & \frac{T\left(e\left(\beta^2+\left(\beta^2+\beta+1\right)T\right)-\beta T-e(\beta-1)T\log\left(\frac{\beta}{\beta-1}\right)\right)}{e\beta^2} \end{aligned} \tag{18}$$

According to (18) and (17), we have

$$\begin{aligned} \rho \geq & \frac{T\left(-\frac{\beta}{\beta-1}+T\log\left(\frac{\beta(T+1)}{(\beta-1)T}\right)+T-1\right)}{\frac{T\left(e\left(\beta^2+(\beta^2+\beta+1)T\right)-\beta T-e(\beta-1)T\log\left(\frac{\beta}{\beta-1}\right)\right)}{e\beta^2}} \\ = & \frac{e\beta^2\left(-\frac{\beta}{\beta-1}+T\log\left(\frac{\beta(T+1)}{(\beta-1)T}\right)+T-1\right)}{e\left(\beta^2+\left(\beta^2+\beta+1\right)T\right)+\beta(-T)-e(\beta-1)T\log\left(\frac{\beta}{\beta-1}\right)}. \end{aligned}$$

Taking $T \to \infty$, it holds that

$$\rho \geq \frac{e\beta^2\left(\log\left(\frac{\beta}{\beta-1}\right)+1\right)}{e\left(\beta^2+\beta+1\right)-\beta-e(\beta-1)\log\left(\frac{\beta}{\beta-1}\right)}.$$

This concludes the proof of the theorem. □

## 6 Conclusion

This paper mainly investigates how the lookahead ability improves the performance of online algorithms for the machine minimization problem, where the jobs have real release times, uniform processing times and equal deadlines. We characterize the competitive ratios, $\frac{e}{\alpha(e-1)+1}$ and $\frac{\beta e}{(\beta-1)e+1}$, as a function of the lookahead ability for the two types of lookahead models, respectively. Corresponding lower bounds are also given. The lookahead variables $\alpha \in [0, 1]$ and $\beta \in [1, +\infty)$ take over all possible levels of lookahead ability, from knowing nothing to knowing full information of the future. As one can see, both of the two ratios decrease in $\alpha$ and $\beta$ respectively. Particularly, the two algorithms are $e$-competitive for $\alpha = 0$ (or $\beta = 1$), which matches the prior result for the case of unit processing times and integer release times and deadlines. When $\alpha = 1$ (or $\beta \to \infty$), the competitive ratio is 1, which means the algorithms output an optimal solution when knowing full information of the future. Regarding future work, it might be interesting to further study a more general version of this problem by dropping some restrictions (e.g. the common deadline assumption). One can also improve the remaining gaps between the upper and lower bounds.

**Acknowledgements** This work was supported by the China Postdoctoral Science Foundation (No. 2020M672646); the National Natural Science Foundation of China (No. 71601152, 71720107002); the National Natural Science Foundation of China-Guangdong Joint Fund (No. U1901223); the Natural Science Foundation of Guangdong Province (No. 2017A030312001); and the Natural Science Basic Research Program of Shaanxi (No. 2020JQ-654).

## References

Allulli L, Ausiello G, Bonifaci V, Laura L (2008) On the power of lookahead in on-line server routing problems. Theor Comput Sci 408(2–3):116–128

Azar Y, Cohen S (2018) An improved algorithm for online machine minimization. Oper Res Lett 46(1):128–133

Bansal N, Kimbrel T, Pruhs K (2007) Speed scaling to manage energy and temperature. J ACM 54(1):3:1–3:39

Chen L, Megow N, Schewior K (2018) An O(log m)-competitive algorithm for online machine minimization. SIAM J Comput 47(6):2057–2077

Devanur NR, Makarychev K, Panigrahi D, Yaroslavtsev G (2014) Online algorithms for machine minimization. CoRR abs/1403.0486

Eidenbenz SJ, Pagourtzis A, Widmayer P (2003) Flexible train rostering. In: Ibaraki T, Katoh N, Ono H (eds) Algorithms and computation, 14th international symposium, ISAAC 2003, Kyoto, Japan, December 15–17, 2003, proceedings, lecture notes in computer science, vol 2906, pp 615–624. Springer, Berlin

Im S, Moseley B, Pruhs K, Stein C (2017) An O(log log m)-competitive algorithm for online machine minimization. In: 2017 IEEE real-time systems symposium, RTSS 2017, Paris, France, December 5–8, 2017, IEEE Computer Society, pp 343–350

Jaillet P, Wagner MR (2008) Generalized online routing: new competitive ratios, resource augmentation, and asymptotic analyses. Oper Res 56(3):745–757

Kao M, Chen J, Rutter I, Wagner D (2012) Competitive design and analysis for machine-minimizing job scheduling problem. In: Chao K, Hsu T, Lee D (eds) Algorithms and computation—23rd international symposium, ISAAC 2012, Taipei, Taiwan, December 19–21, 2012. Proceedings, lecture notes in computer science, vol 7676, pp 75–84. Springer, Berlin

Li W, Yuan J, Cao J, Bu H (2009) Online scheduling of unit length jobs on a batching machine to maximize the number of early jobs with lookahead. Theor Comput Sci 410(47–49):5182–5187

Mandelbaum M, Shabtay D (2011) Scheduling unit length jobs on parallel machines with lookahead information. J Sched 14(4):335–350

Phillips CA, Stein C, Torng E, Wein J (2002) Optimal time-critical scheduling via resource augmentation. Algorithmica 32(2):163–200

Saha B (2013) Renting a cloud. In: Seth A, Vishnoi NK (eds) IARCS annual conference on foundations of software technology and theoretical computer science, FSTTCS 2013, December 12–14, 2013, Guwahati, India, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, LIPIcs, vol 24, pp 437–448

Zhang H, Tong W, Xu Y, Lin G (2016) The steiner traveling salesman problem with online advanced edge blockages. Comput Oper Res 70:26–38

Zheng F, Xu Y, Zhang E (2008) How much can lookahead help in online single machine scheduling. Inf Process Lett 106(2):70–74

Zheng F, Cheng Y, Liu M, Xu Y (2013) Online interval scheduling on a single machine with finite lookahead. Comput Oper Res 40(1):180–191

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.