A new average case analysis for completion time scheduling

Mark Scharbrodt
scharbro@in.tum.de

Thomas Schickinger
schickin@in.tum.de

Angelika Steger
steger@in.tum.de

Institut für Informatik
Technische Universität München
D–80290 München, Germany

ABSTRACT

We present a new average case analysis for the problem of scheduling n jobs on m machines so that the sum of job completion times is minimized. Our analysis transfers the concept of competitive analysis — which is a typical worst case notion — to the average case. We show that the classic SEPT scheduling strategy with $\Omega(n)$ worst case competitive ratio achieves $\mathcal{O}(1)$ on the average. Moreover, bounds on the probability distribution of the competitive ratio are derived which provide an in-depth understanding of the stochastic version of the min sum scheduling problem.

1. INTRODUCTION

In online scheduling the competitive ratio has become a standard approach for evaluating the performance of a given algorithm. The competitive ratio, however, is a worst case notion and usually the gap between the optimal solution and the solution obtained by the online algorithm is considerably smaller. For a closer look at the typical behavior of an algorithm we thus need average case results.

In our paper we merge competitive analysis known from online scheduling with the average case analysis for stochastic problems. We interpret the competitive ratio itself as a random variable and compute its value in expectation. The specific problem we consider is that of scheduling n jobs on m machines with the aim of minimizing the total (unweighted) sum of job completion times. Here job processing times are assumed to be exponentially or gamma distributed. We compute the expected value $\mathbb{E}\left[V^{\mathrm{Alg}}/V^{\mathrm{Opt}}\right]$, where V^{Alg} corresponds to the value of the objective function achieved by a certain algorithm and V^{Opt} denotes the value achieved by an optimal offline algorithm.

Previous work

A first motivation for our paper comes from successful average case analysis for bin packing problems. For instance in [3] and [8] the asymptotic average performance of the classic Next Fit algorithm has been computed. In stochastic scheduling a similar approach has been used by Coffman and Gilbert in [2]. For the minimum makespan problem they gave bounds on the expected ratio for the LEPT scheduling algorithm and estimated the tails of the corresponding distribution.

However, this approach is not at all common in this field, even though there is a substantial literature on stochastic scheduling (see e.g. [5] [19] [12] [13] [11] [18] [7] [15] [9]). Generally speaking, the probabilistic analysis of scheduling problems tends to be technically hard and often involves non–combinatorial arguments. Typical average case results examine the ratio $\mathbb{E}[V^{\mathrm{Alg}}]/\mathbb{E}[V^{\mathrm{online}}]$ between the average performance of a given algorithm and the average performance of the best possible *online* algorithm. Optimal algorithms are only known for few scheduling problems. For the minimum makespan problem for instance, the LEPT rule (*longest expected processing time first*) is optimal with respect to the expected value of the objective function (see e.g. [7]). The opposite SEPT (*shortest expected processing time first*) list scheduling algorithm minimizes the unweighted sum of completion times on parallel machines [16]. For the weighted version of the problem the approach in [14] uses polyhedral relaxations in order to derive constant factor performance guarantees for $\mathbb{E}[V^{\mathrm{Alg}}]/\mathbb{E}[V^{\mathrm{online}}]$ for priority rules which are guided by optimal LP solutions.

Competitive ratio and average case analysis

The success of competitive analysis in the last years sheds new light on the approach presented in [2]. We believe that $\mathbb{E}[V^{\mathrm{Alg}}/V^{\mathrm{Opt}}]$, which we call *expected competitive ratio*, represents an attractive alternative to the classic performance measure $\mathbb{E}[V^{\mathrm{Alg}}]/\mathbb{E}[V^{\mathrm{online}}]$. Firstly, note that $V^{\mathrm{Alg}}/V^{\mathrm{Opt}}$ exceeds $\mathbb{E}[V^{\mathrm{Alg}}/V^{\mathrm{Opt}}]$ by more than a factor of t with probability at most $1/t$ due to Markov's inequality. Hence, with high probability the expected competitive ratio shows the right order of magnitude of the competitive ratio.

Secondly, the value of $\mathbb{E}[V^{\mathrm{Alg}}]$ can be dominated by a fraction of instances where the objective function is large. The behavior of the algorithm on the remaining 'simpler' instances is not taken into account. With $\mathbb{E}[V^{\mathrm{Alg}}/V^{\mathrm{Opt}}]$, however, all instances are weighted with V^{Opt} and have, thus, the same impact on the performance measure.

Finally, results on the expected competitive ratio can directly be compared to well known worst case bounds on

STOC'02, May 19-21, 2002, Montreal, Quebec, Canada.

the competitive ratio. The problem considered in this paper shows that the expected competitive ratio may be much smaller than the worst case ratio.

Our results

In this paper we consider the problem of minimizing the sum of completion times for parallel machines and prove that the expected competitive ratio is of order $\mathcal{O}(1)$ for the SEPT list scheduling algorithm if the processing times are exponentially distributed. This result is extended to the gamma distribution, which comprises the exponential distribution and the Erlang distribution as special cases.

On the other hand it is easily seen that the competitive ratio may be as large as $\Omega(n)$ in the worst case. This shows that a worst case analysis of the competitive ratio may yield results which are much too pessimistic.

When relating results on the expected competitive ratio to results on $\mathbb{E}[V^{\text{Alg}}]/\mathbb{E}[V^{\text{online}}]$ one should note that in our approach an algorithm is compared to a stronger opponent, namely an optimal *offline* algorithm. This again raises the question about optimality of a scheduling strategy. In particular we show that the SEPT strategy is *not* generally optimal for the unweighted version of min sum scheduling. It turns out that there are examples where this strategy has a poorer competitive performance than the opposite LEPT strategy *in expectation.* This contrasts to the common view on the SEPT strategy, which is said to be optimal for (unweighted) min sum scheduling.

Remarks on the analysis

The analysis of the random variable $V^{\text{Alg}}/V^{\text{Opt}}$ is complicated by the fact that standard techniques which exploit the linearity of expectation cannot be applied. To cope with that problem we present a technique where the probability space is partitioned according to a series of 'bad events' which are defined by means of the deviation of the individual jobs from their expected processing time. Using these events we will show that $V^{\text{Alg}}/V^{\text{Opt}} = \Omega(\ln^{(l)}(n))$ with probability at most $\mathcal{O}((\ln^{(l)}(n))^{-2})$ for $l = 0 \dots \ln^*(n)$. This suffices to achieve an overall constant expected performance.

Since this proof strategy does not rely on problem specific structures we believe that our approach is generally applicable for proving bounds on the expected competitive ratio. As a byproduct this approach yields bounds on the probability that a schedule computed by the given algorithm differs significantly from the optimum, thereby providing a rough estimate of the distribution of $V^{\text{Alg}}/V^{\text{Opt}}$. In that way we explore the transition from the worst case competitive ratio to the expected competitive ratio.

All in all, we believe that the expected competitive ratio represents a useful measure for average case analysis with many desirable properties. Although a direct analysis of the random variable $V^{\text{Alg}}/V^{\text{Opt}}$ may not be tractable in many applications we feel that these difficulties can be overcome by developing suitable proof techniques and hope that this paper will serve as a starting point for further research in this field.

Outline of the Paper

This paper is organized as follows: In Section 2 we introduce our stochastic scheduling model and give an example which shows that the SEPT scheduling rule does not minimize the expected competitive ratio in general. In Section 3 we consider the case of exponentially distributed job processing times. Here, we first prove the optimality of the SEPT scheduling rule and give an $\mathcal{O}(1)$ bound for the expected competitive ratio. We then extend this result to a more general class of distributions and to the parallel machine case.

2. MODEL AND NOTATION

We closely follow the terminology and definitions used in [12, 14]. Throughout the paper we consider the scheduling problem $P \mid p_j \sim stoch \mid \sum C_j$. This scheduling problem is given by a finite set of jobs $J = \{1, \cdots, n\}$ that have to be processed non-preemptively so as to minimize the total sum of completion times $\sum_{j \in J} C_j$, where C_j denotes the completion time of job j. The duration of each given job i is described by a random variable p_i.

Jobs are scheduled *online* according to a given strategy Π. A strategy Π can be identified with a function that assigns to each possible outcome $(p_1, \dots, p_n)$ of the random processing times a vector $(C_1^\Pi, \dots, C_n^\Pi)$ of job completion times achieved under strategy Π. This defines a random variable $\sum_{j \in J} C_j^\Pi$ for the value of the objective function.

Given the outcome $p = (p_1, \dots, p_n)$ of the processing times one can also identify a corresponding optimal (offline) schedule for the jobs and assign its value to p. This defines a random variable $\sum_{j \in J} C_j^{\text{Opt}}$, where C_j^{Opt} denotes the completion time of job j under a (not necessarily polynomial time) optimal offline algorithm. Then the ratio

$$\frac{\sum_{j \in J} C_j^\Pi}{\sum_{j \in J} C_j^{\text{Opt}}}$$

is a random variable that describes the competitive ratio of a strategy Π compared to the best possible schedule that could be achieved if the processing times were all known in advance. This relative performance ratio can be interpreted as the probabilistic version of the *competitive ratio* used in online scheduling.

Definition 1. *A strategy is said to be* competitive optimal *for $P \mid p_j \sim stoch \mid \sum C_j$ if it minimizes the expected competitive ratio, i.e., if it yields the infimum over the set*

$$\left\{ \mathbb{E}\left[\frac{\sum_j C_j^\Pi}{\sum_j C_j^{\text{Opt}}}\right] \mid \Pi \ \textit{strategy} \right\}$$

Note that we defined the optimality of a strategy with regard to our given scheduling problem $P \mid p_j \sim stoch \mid \sum C_j$, but the concept easily transfers to arbitrary stochastic optimization problems.

We give a simple example in order to illustrate the difference between minimizing the expected value of the objective function and the expected competitive ratio. We consider the problem $1 \mid p_j \sim stoch \mid \sum C_j$ where the aim is to minimize the sum of completion times among the jobs on a single machine. In the deterministic version of the problem an optimal schedule is obtained by ordering the jobs according to non–decreasing processing times (*Smith's rule*) [17, 10]. For the stochastic variant of the problem the rule transfers to the *shortest expected processing time first rule* (SEPT). SEPT simply schedules the jobs according to non-decreasing expected processing times. In [16], SEPT has been shown to be optimal in the sense that it minimizes the expected sum of completion times. Consider, however, the the following instance of our scheduling problem: Job one has a deterministic processing time of $p_1 = 1/2$ and job two has processing time $p_2 = 2$ with probability $\Pr[p_2 = 2] = 1/2$ and processing time $p_2 = \epsilon$ for a sufficiently small ϵ with probability $\Pr[p_2 = \epsilon] = 1/2$. For $\epsilon \to 0$ one directly computes that the SEPT strategy has an expected competitive ratio 3/2 which is larger than that of the opposite LEPT-strategy. We hence deduce that the classic performance measure which considers SEPT as the optimal scheduling strategy for $P \mid p_j \sim stoch \mid \sum C_j$ might be misleading, since SEPT can have a rather poor relative performance even *in expectation.*

3. EXPECTED COMPETITIVE RATIO FOR EXPONENTIALLY DISTRIBUTED JOB PROCESSING TIMES

In this section we will consider the single machine problem and moreover, we assume that the job processing times are exponentially distributed.

Our first results states that the *SEPT*–scheduling rule is *competitive optimal* for exponentially distributed job processing times. The proof uses classic exchange arguments.

THEOREM 2. *If the processing times p_j are exponentially distributed with parameter λ_j, then a strategy Π is competitive optimal for $1 \mid p_j \sim exp(\lambda_j) \mid \sum C_j$, if and only if Π is SEPT.*

We next compute the expected competitive ratio for identically distributed job processing times. As one easily observes, the worst case ratio between the solution generated under SEPT and the optimal offline solution is equal to n. The expected competitive ratio however is *independent* of the number of jobs:

THEOREM 3. *The expected competitive ratio $\mathbb{E}\left[\frac{\sum_j C_j^{\text{SEPT}}}{\sum_j C_j^{\text{Opt}}}\right]$ of the SEPT–scheduling rule is bounded by a constant less than 3 for any given number $n > 3$ of jobs if all jobs have exponentially distributed processing times with identical mean.*

PROOF. For simplicity of notation we assume that all jobs have expected processing time 1, i.e. they all have density function $f(x) = e^{-x}$. Our proof can be easily generalized to the case, where all density functions share the same arbitrary parameter $\lambda > 0$, i.e., $f_i(x) = \lambda e^{-\lambda x}$ for all jobs $i = 1, \ldots, n$.

The random vector $p = (p_1, \ldots, p_n)$ of the job processing times induces a family of (dependent) random variables $S_1, \cdots, S_n$, where S_i denotes the processing time of the i^{th} shortest job. Standard properties of the exponential distribution yield $\mathbb{E}[S_i] = \sum_{k=1}^{i} \frac{1}{(n+1-k)}$. We can also write $S_1 = Y_1'$, $S_2 = Y_1' + Y_2'$ and generally $S_i = \sum_{j=1}^{i} Y_j' = \sum_{j=1}^{i} \frac{1}{n+1-j} Y_j$. Here, $Y_1', \ldots, Y_n'$ denote *independent* random variables with exponential density functions with parameter $(n+1-j)$. $Y_1, \ldots, Y_n$ denote random variables with standard exponential distribution.

A schedule of the jobs on a single machine is given by a permutation π of the jobs. For a scheduling strategy *Alg* we have

$$\sum_{j\in J} C_j^{\text{Alg}} = \sum_{j=1}^{n} (n + 1 - \pi^{\text{Alg}}(j)),$$

where $\pi^{\text{Alg}}(j)$ denotes the position of j^{th} shortest job in the schedule/permutation calculated by *Alg*. Setting $\pi^{\text{Opt}}(j) = j$ and $\pi^{\text{SEPT}}(j) =: \pi_j$ we obtain

$$\frac{\sum_{j\in J} C_j^{\text{SEPT}} - \sum_{j\in J} C_j^{\text{Opt}}}{\sum_{j\in J} C_j^{\text{Opt}}} = \frac{\sum_{i=1}^{n}(i-\pi_i)S_i}{\sum_{i=1}^{n}(n+1-i)S_i}.$$

As all jobs have the same expected processing time each ordering π of the jobs is equally likely under the SEPT scheduling rule. In order to compute the expected competitive ratio, we therefore have to consider all $n!$ possible permutations of the jobs. Let $s = (s_1, \ldots, s_n) \subseteq \mathbb{R}_+^n$ denote a specific outcome of the random vector $S = (S_1, \ldots, S_n)$ with product density function $f_n(s)$, $s \in \mathbb{R}_+^n$. Then,

$$\begin{aligned}
\mu := \mathbb{E}\left[\frac{\sum_{j\in J} C_j^{\text{SEPT}} - \sum_{j\in J} C_j^{\text{Opt}}}{\sum_{j\in J} C_j^{\text{Opt}}}\right] & \\
= \sum_{\pi} \frac{1}{n!} \mathbb{E}\left[\frac{\sum_{i=1}^{n}(i-\pi_i)S_i}{\sum_{i=1}^{n}(n+1-i)S_i}\right] & \\
= \int_s f_n(s) \sum_{\pi} \frac{1}{n!} \frac{\sum_{i=1}^{n}(i-\pi_i)s_i}{\sum_{i=1}^{n}(n+1-i)s_i}\, ds. &
\end{aligned}$$

Now we use

$$\frac{1}{n!}\sum_{\pi}\sum_{i=1}^{n} \pi_i s_i = \sum_{i=1}^{n} s_i \cdot \frac{1}{n!}(n-1)! \sum_{j=1}^{n} j = \sum_{i=1}^{n} s_i \cdot \frac{n+1}{2},$$

and obtain

$$\begin{aligned}
\mu &= \int_s f_n(s) \frac{\sum_{i=1}^{n}(i-\frac{n+1}{2})s_i}{\sum_{i=1}^{n}(n+1-i)s_i}\, ds \\
&= \mathbb{E}\left[\frac{\sum_{i=1}^{n}(i-\frac{n+1}{2})S_i}{\sum_{i=1}^{n}(n+1-i)S_i}\right] \\
&= \mathbb{E}\left[\frac{\sum_{i=1}^{n}\sum_{j=1}^{i}(i-\frac{n+1}{2})Y_j'}{\sum_{i=1}^{n}\sum_{j=1}^{i}(n+1-i)Y_j'}\right] \\
&= \mathbb{E}\left[\frac{\sum_{j=1}^{n}\sum_{i=j}^{n}(i-\frac{n+1}{2})Y_j'}{\sum_{j=1}^{n}\sum_{i=j}^{n}(n+1-i)Y_j'}\right] = \mathbb{E}\left[\frac{\sum_{j=1}^{n} c_{n,j}Y_j}{\sum_{j=1}^{n} d_{n,j}Y_j}\right],
\end{aligned}$$

where

$$c_{n,j} = \sum_{i=j}^{n}\left(i - \frac{n+1}{2}\right)\frac{1}{n+1-j} = \frac{j-1}{2},$$

$$d_{n,j} = \sum_{i=j}^{n}(n+1-i)\frac{1}{n+1-j} = \frac{n+2-j}{2}.$$

For simplicity of notation we set $c_j := c_{n,j}$ and $d_j := d_{n,j}$. Note, that $d_j > d_{j+1}$ for all $j = 1, \ldots, n-1$.

In the sequel we will need two simple properties of the exponential distribution. If $X_1, \ldots, X_n$, $n \geq 2$ are mutually independent random variables with the standard exponential distribution with mean 1, then

$$\mathbb{E}\left[\frac{X_i}{X_1 + \cdots + X_n}\right] = \frac{1}{n}, \tag{3}$$

for all $i = 1, \ldots, n$, and

$$\mathbb{E}\left[\frac{1}{X_1 + \cdots + X_n}\right] = \frac{1}{n-1}. \tag{4}$$

Using (3) and (4) we obtain (without loss of generality we assume that n is even)

$$\begin{aligned}
\mu &= \sum_{j=1}^{n} c_j \mathbb{E}\left[\frac{Y_j}{\sum_{i=1}^{n} d_i Y_i}\right] < \sum_{j=1}^{n} c_j \mathbb{E}\left[\frac{Y_j}{\sum_{i=1}^{\frac{n}{2}} d_{\frac{n}{2}} Y_i}\right] \\
&= \sum_{j=1}^{n} \frac{c_j}{d_{\frac{n}{2}}} \mathbb{E}\left[\frac{Y_j}{\sum_{i=1}^{\frac{n}{2}} Y_i}\right] \\
&= \sum_{j=1}^{\frac{n}{2}} \frac{c_j}{d_{\frac{n}{2}}} \mathbb{E}\left[\frac{Y_j}{\sum_{i=1}^{\frac{n}{2}} Y_i}\right] + \sum_{j=\frac{n}{2}+1}^{n} \frac{c_j}{d_{\frac{n}{2}}} \mathbb{E}[Y_j] \mathbb{E}\left[\frac{1}{\sum_{i=1}^{\frac{n}{2}} Y_i}\right] \\
&= \sum_{j=1}^{\frac{n}{2}} \frac{c_j}{d_{\frac{n}{2}}} \frac{1}{\frac{n}{2}} + \sum_{j=\frac{n}{2}+1}^{n} \frac{c_j}{d_{\frac{n}{2}}} \frac{1}{\frac{n}{2}-1} \\
&= \sum_{j=1}^{n} \frac{c_j}{d_{\frac{n}{2}}} \frac{1}{\frac{n}{2}-1} - \sum_{j=1}^{\frac{n}{2}} \frac{c_j}{d_{\frac{n}{2}}} \frac{1}{\frac{n}{2}(\frac{n}{2}-1)} \\
&= \sum_{j=1}^{n} \frac{j-1}{(\frac{n}{2}+2)(\frac{n}{2}-1)} - \sum_{j=1}^{\frac{n}{2}} \frac{j-1}{(\frac{n}{2}+2)\frac{n}{2}(\frac{n}{2}-1)} \\
&= 2 \cdot \frac{2n^2-3n+2}{n^2+2n-8} < 2,
\end{aligned}$$

for $n \geq 3$. Thus our theorem is proved. □

3.1 Expected competitive ratio in the general case

In this section we generalize our analysis and allow exponential distributions with variable parameters. Thus the density function f_i for job i can be written as $f_i(x) = \lambda_i e^{-\lambda_i x}$, with arbitrary $\lambda_i \geq 0$. Our results can be easily transferred however, to the parallel machine case and to more general distributions. Again we will prove that a constant expected competitive ratio is achieved by the SEPT scheduling algorithm.

As the arguments used in the proof of Theorem 3 are not valid for general parameters λ_i we take a completely different approach for the analysis of the general case. Before going into technical details we will try to explain the intuition behind the ideas of our proof.

Our basic idea is to balance jobs j with an extraordinarily high contribution C_j^{SEPT} by other jobs with a 'reasonable' contribution in order to compare $\sum_j C_j^{\text{SEPT}}$ to $\sum_j C_j^{\text{Opt}}$. Note that the contribution of a job j to the objective function is given by $C_j^{\text{Opt}} = w_j^{\text{Opt}} \cdot p_j$, where w_j^{Opt} denotes the number of jobs which are scheduled after j (including j) in the optimal solution. The contribution to the SEPT-solution amounts to $w_j^{\text{SEPT}} := (n-j+1)p_j$, where we assume without loss of generality that the jobs $1,\ldots,n$ are already in SEPT-order.

In a first step we modify the problem by reducing the processing times of jobs j which are larger than, say, $\nu\mu_j$ for some $\nu > 0$, where $\mu_j := \mathbb{E}[p_j]$. The number of jobs affected by this modification can be estimated precisely. Moreover, the order in which the jobs with reduced processing times occur in the optimal solution is identical to the SEPT-order. This relates w_j^{Opt} to w_j^{SEPT}.

Of course, this approach introduces a new inaccuracy as we have to exchange the real processing times with their reduced counterparts. The multiplicative error due to that directly corresponds to the multiplicative deviation of the processing times from their expectation. If the largest deviation is given by a factor of, say, $\mathcal{O}(\ln(n))$ we can immediately deduce a bound of $\mathcal{O}(\ln(n))$ on the expected competitive ratio.

However, we would like to improve on that. Note that a single job with a large deviation does not yet destroy the whole competitive ratio. Let c_j denote the expected contribution of job j to the objective function of the SEPT-rule and assume that its actual contribution is given by δc_j. If there are δ jobs with a contribution of the same order of magnitude as c_j then the factor δ in δc_j has no significant influence on the order of magnitude of the objective function. Hence, we may balance a job with a multiplicative deviation of δ by δ jobs of reasonable size. Furthermore, we require that these jobs are 'near' to j in the schedule, i.e., that they have a weight similar to w_j in the objective function. Finally, we have to show that the sets which are used to compensate for the 'extraordinarily' large jobs can be chosen to be disjoint.

This approach bears one major technical problem, though. If there are many jobs with significant deviation from the expected processing time or if such jobs occur in unusual patterns we can only balance them to a limited extent. Therefore, we suitably partition the probability space and estimate $V^{\text{Alg}}/V^{\text{Opt}}$ successively. As a byproduct we obtain a family of bounds on the distribution of $V^{\text{Alg}}/V^{\text{Opt}}$, namely we estimate the probability that $V^{\text{Alg}}/V^{\text{Opt}}$ exceeds a certain order of magnitude. This leads to a better understanding of the performance of the SEPT-rule than the value of $\mathbb{E}[V^{\text{Alg}}/V^{\text{Opt}}]$ alone.

3.1.1 Definitions

We define $\ln^*(n) := \min\{i \in \mathbb{N} \mid \ln^{(i)}(n) \leq C\}$ for a sufficiently large constant $C \geq 1$. Note that this definition slightly differs from the standard definition (where $\ln^*(n)$ is defined with respect to $C = 1$).

Definition 1. A continuous random variable X with mean $\mathbb{E}[X] = \mu$ is said to satisfy *Property* $\mathcal{P}(\alpha,\nu)$ if $\Pr[X \geq \nu\mu] \geq 1/2$ and there is a $z_0 > 0$ such that $\Pr[X \geq \alpha\ln(z)\mu] \leq z^{-5}$ for all $z \geq z_0$.

As we will show in the sequel, Property $\mathcal{P}(\alpha,\nu)$ is the only requirement on the distribution of the processing times that is needed to prove a constant expected competitive ratio for the SEPT-strategy. Note that an exponentially distributed random variable X with $\mathbb{E}[X] = \mu$ satisfies $\mathcal{P}(\alpha,\nu)$ for $\alpha = 5$ and $\nu = \ln(2)$, since $\Pr[X \geq t] = e^{-t/\mu}$ for $t \geq 0$. Using the constants $\alpha = 5$, $\nu = \ln(2)$, $L := \ln^*(n)$, $c = 360$ and $C \geq 1$ sufficiently large, so that all inequalities below hold, we define

$$\begin{aligned}
S_1 &:= \{j \mid p_j \geq \alpha\ln(n)\mu_j\}, \\
S_l &:= \{j \mid p_j \geq \alpha\ln^{(l)}(n)\mu_j \ \wedge\ p_j < \alpha\ln^{(l-1)}(n)\mu_j\} \\
&\qquad \text{for } l = 2,\ldots,L, \\
T &:= \{j \mid p_j \geq \nu\mu_j\}, \\
R &:= \{j \mid p_j < \nu\mu_j\},
\end{aligned}$$

where $\mu_j := \mathbb{E}[p_j]$. Furthermore, we introduce the abbreviations $[x] := \{1,\ldots,x\}$, $S_{\geq k} := \bigcup_{l=k}^{L} S_l$ (resp. $S_{\leq k} := \bigcup_{l=1}^{k} S_l$) and $S_k[x] := S_k \cap [x]$ for $k = 2,\ldots,L$. $T[x]$, $R[x]$ and $S_{\geq k}[x]$ are defined analogously.

For $l = 2,\ldots,L-1$ we also partition T into (consecutive) groups of size $(\ln^{(l)}(n))^3$. In order to define this partition let $t(j) := |T \cap \{1,\dots,j\}|$ denote the number of jobs up to j which belong to the set T. Then

$$T_{l,i} := \{j \in T \mid (i-1)\cdot(\ln^{(l)}(n))^3+1 \le t(j) \le i\cdot(\ln^{(l)}(n))^3\}.$$

A group $T_{l,i}$ is called *dense* if it contains 'many' jobs from S_{l+1}. More precisely, for $l = 1,\dots,L-1$ and arbitrary i a set $T_{l,i}$ is called *dense*, if $|T_{l,i} \cap S_{l+1}| \ge \ln^{(l)}(n)$. $T_{L,i}$ is defined not to be dense. Furthermore, we let

$$D_l := \{\min(T_{l,i} \cap S_{l+1}) \text{ for all dense groups } T_{l,i}\}$$

and define a set C_l for $l = 1,\dots,L$ as follows:

$$C_l := S_l \cup D_l.$$

Informally speaking, C_l contains all jobs with a critically large multiplicative deviation from their expected processing time. Suppose that we want to show a bound of order $\ln^{(l)}(n)$ on the expected competitive ratio. Then the jobs in C_l will play in important rôle in our proof since their deviation from their expected processing time is of the same order as the desired bound. A job in C_l is either comparatively large by itself (i.e., it belongs to S_l) or it is the first job in S_{l+1} of a dense group $T_{l,i}$. Note that such a dense group contains many jobs from S_{l+1} which must also be considered very carefully, since they deviate significantly from their expectation (although not as much as the jobs in S_l) and occur in a big 'bundle'.

Next we define a sequence of 'bad events' $\bar{\mathcal{A}}_l,\ l = 2,\dots,L$ which are shown to occur with small probability. On the other hand, if they do not occur we get a bound on the expected competitive ratio, as we will prove later. Let

$$\bar{\mathcal{A}}_1 \ :\ \exists j \in J \ :\ j \in S_1.$$

For $l = 2,\dots,L$ we let $\bar{\mathcal{A}}_l = \bar{\mathcal{A}}_{l,1} \cup \bar{\mathcal{A}}_{l,2} \cup \bar{\mathcal{A}}_{l,3}$, where

$$\begin{aligned}
\bar{\mathcal{A}}_{l,1} &: \ \exists k :\ |C_l \cap \{n-15ck(\ln^{(l-1)}(n))^2+1,\dots,n\}| \ge k,\\
\bar{\mathcal{A}}_{l,2} &: \ \exists i \in \{1,\dots,n-c\ln^{(l)}(n)\} \ :\\
&\qquad |T \cap \{i+1,\dots,n\}| \le \tfrac{1}{4}(n-i),\\
\bar{\mathcal{A}}_{l,3} &: \ \exists i \in \{1,\dots,n-c\ln^{(l)}(n)\} \ :\\
&\qquad |T \cap \{i+1,\dots,i+\tfrac{n-i}{5}\}| \le \tfrac{n-i}{15}.
\end{aligned}$$

The complement of a bad event $\bar{\mathcal{A}}_l$ is denoted by $\mathcal{A}_l := \Omega \setminus \bar{\mathcal{A}}_l$.

To show that the probability of the bad events defined above is comparatively small, we only use the relatively weak property $\mathcal{P}(\alpha,\nu)$ of the processing times given in Definition 1. Using tail bounds for the binomial distribution this suffices to bound the probability of the bad events.

Lemma 4. *For all $l = 1,\dots,L$ we have that $\Pr[\bar{\mathcal{A}}_l] = \mathcal{O}\left((\ln^{(l-1)}(n))^{-2}\right)$, where the constants hidden in the $\mathcal{O}$–notation are independent of l.*

Proof (Sketch). It suffices to show that

$$\Pr[\bar{\mathcal{A}}_{l,k}] = \mathcal{O}\left((\ln^{(l-1)}(n))^{-2}\right)$$

for $k = 1,2,3$. Here we will concentrate on the case $k = 2$. The other two cases follow similarly.

Note that Definition 1 directly implies that $\Pr[j \in T] \ge 1/2$ for all $j \in J$. Let E_i denote the event that $|T \cap \{i+1,\dots,n\}| \le \frac{1}{4}(n-i)$. Using the Chernoff-bound

$$\Pr[\mathrm{Bin}(n,p) \le (1-\delta)np] \le e^{-\delta^2 np/2}$$

we deduce that

$$\Pr[E_i] \le \Pr[\mathrm{Bin}(n-i,1/2) \le (n-i)/4] \le e^{-(n-i)/16}.$$

It follows that

$$\begin{aligned}
\Pr[\bar{\mathcal{A}}_{l,2}] &\le \sum_{i=1}^{n-c\ln^{(l)}(n)} \Pr[E_i] \le \sum_{i=c\ln^{(l)}(n)}^{n-1} e^{-i/16}\\
&\le e^{-c\ln^{(l)}(n)/16}\cdot\sum_{i=0}^{\infty} e^{-i/16}\\
&= \mathcal{O}((\ln^{(l-1)}(n))^{-c/16}) = \mathcal{O}((\ln^{(l-1)}(n))^{-2}),
\end{aligned}$$

by choice of c. $\square$

3.1.2 Calculating the Expected Ratio

In this section we will show that the SEPT rule achieves constant expected competitive ratio. In order to do so we will heavily make use of the 'bad events' $\bar{\mathcal{A}}_l$. In fact, as we will show in Lemma 7 below, they are defined in such a way that if none of the events $\bar{\mathcal{A}}_1,\dots,\bar{\mathcal{A}}_l$ holds, then we can bound the competitive ratio *deterministically* by $\mathcal{O}(\ln^{(l)}(n))$. For the proof of that lemma we need two auxiliary lemmas which we now state.

Lemma 5. *Let $l \in \{2,\dots,L\}$ be arbitrary but fixed and assume that the event $\mathcal{A}_l$ holds. Then there exists for $j \in C_l$ a set $A_j \subseteq [n]$ such that the following conditions are satisfied:*

1. $|A_j| = (\ln^{(l-1)}(n))^2$,
2. $A_j \subseteq T[n-c\ln^{(l)}(n)] \cap \{j,\dots,j+\frac{n-j}{5}\}$,
3. $\max A_j < \min A_{j'}$ *for all* $j,j' \in C_l$ *with* $j < j'$, *and either* $\max A_j < i$ *or* $i < \min A_j$ *for all* $i \in T \setminus A_j$.

Proof. We construct the sets A_j using a greedy type algorithm. More precisely, we iterate through $j \in C_l$ (in the order of increasing j's) and greedily allocate disjoint sets $A_j \subseteq T[n-c\ln^{(l)}(n)] \cap \{j,\dots,j+\frac{n-j}{5}\}$, always taking the jobs with smallest numbers of those which are eligible.

We need to show that this process cannot fail. Assume the contrary. Then there must be jobs $i_1,\dots,i_k \in C_l$ such that the sets $A_{i_1},\dots,A_{i_{k-1}}$ occupy so many jobs that A_{i_k} does not fit within $T[n-c\ln^{(l)}(n)] \cap \{i_k,\dots,i_k+\frac{n-i_k}{5}\}$. Note that we may assume without loss of generality that either $i_1 = 1$ or $\max A_{i_1-1} < i_1$. Hence, the fact that the algorithm fails can be rewritten as follows:

$$\begin{aligned}
\sum_{s=1}^{k}|A_{i_s}| &= k\cdot(\ln^{(l-1)}(n))^2\\
&> |\ T[n-c\ln^{(l)}(n)] \cap \{i_1,\dots,i_k+\tfrac{n-i_k}{5}\}\ |\\
&\ge |\ T[n-c\ln^{(l)}(n)] \cap \{i_1,\dots,i_1+\tfrac{n-i_1}{5}\}\ |.
\end{aligned} \tag{5}$$

As we conditioned on the event $\mathcal{A}_l$ we deduce from the definition of $\bar{\mathcal{A}}_{l,1}$ (setting $k = 1$) that

$$C_l \cap \{n-15c(\ln^{(l-1)}(n))^2+1,\dots,n\} = \emptyset, \tag{6}$$

and from $\bar{\mathcal{A}}_{l,3}$ that

$$|\,T \cap \{i+1,\dots,i+\tfrac{n-i}{5}\}\,| \ \ge\ \tfrac{n-i}{15} \tag{7}$$

for all $i = 1,\dots,n-c\ln^{(l)}(n)$.

Due to (6) we have $j \le n - 15c(\ln^{(l-1)}(n))^2$ for all $j \in C_l$ and, thus, $j + \frac{n-j}{5} \le n - c\ln^{(l)}(n)$. It follows that

$$\begin{aligned} |\,T[n - c\ln^{(l)}(n)] \quad &\cap \quad \{j,\ldots,j+\tfrac{n-j}{5}\}\,| \\ &= \quad |\,T \cap \{j,\ldots,j+\tfrac{n-j}{5}\}\,| \\ &\overset{(7)}{\ge} \quad \tfrac{n-j}{15}. \end{aligned} \tag{8}$$

Combining (5) and (8) we deduce that

$$k \cdot (\ln^{(l-1)}(n))^2 > \tfrac{n-i_1}{15},$$

which is equivalent to

$$i_1 > n - 15k(\ln^{(l-1)}(n))^2 \ge n - 15ck(\ln^{(l-1)}(n))^2.$$

Thus, $\{n - 15ck(\ln^{(l-1)}(n))^2 + 1,\ldots,n\}$ contains $i_1,\ldots,i_k \in C_l$, contradicting the assumption that $\mathcal{A}_l$ holds. □

LEMMA 6. *Let $l^* \in \{2,\ldots,L\}$ be fixed and assume that the events $\mathcal{A}_1,\ldots,\mathcal{A}_{l^*}$ hold. Then we can find pairwise disjoint sets B_j for $j \in \bigcup_{l=1}^{l^*} S_l$ such that the following conditions are satisfied:*

1. $|B_j| = \alpha\ln^{(l-1)}(n)$ *if* $j \in S_l$,
2. $B_j \subseteq T[n - c\ln^{(l)}(n)] \cap \{j,\ldots,j+\frac{n-j}{5}\}$.

PROOF. The main idea is to use the sets A_j from Lemma 5 in order to construct the sets B_j iteratively. We start with $l = l^*$ and then reduce l subsequently to $l = 2$. For each l we define sets B_j for two kinds of jobs j. Namely, those jobs j which are contained in S_l and *not* in a dense group $T_{l-1,i}$ and those jobs j which are contained in S_{l+1} *and* some dense group $T_{l,i}$. In other words, for $l = 2,\ldots,l^*$ we assign the sets B_j for all $j \in X_l$, where

$$X_l := (S_l \setminus \bigcup_{i\,:\,T_{l-1,i}\text{ dense}} T_{l-1,i}) \;\cup\; (S_{l+1} \cap \bigcup_{i\,:\,T_{l,i}\text{ dense}} T_{l,i}),$$

and, with abuse of notation, $T_{0,i} := \emptyset$. (Note that $\bigcup_{l=2}^{l^*} X_l \supseteq \bigcup_{l=2}^{l^*} S_l = \bigcup_{l=1}^{l^*} S_l$, since $S_1 = \emptyset$ if $\mathcal{A}_1$ occurs.) The assignment of the sets B_j is done by the following algorithm:

```
B_t := ∅ for all t ∈ ⋃_l S_l;
for l from l* downto 2 do
  let A_j denote the sets from Lemma 5 for all j ∈ C_l;
  forall j ∈ S_l ∩ X_l do
    assign to B_j the smallest α ln^(l-1)(n) elements
    from A_j \ ⋃_t B_t;
  forall j ∈ D_l do
    let i denote the dense group T_{l,i} such
    that j = min(T_{l,i} ∩ S_{l+1}).
    forall j' ∈ T_{l,i} ∩ S_{l+1} do
      assign to B_{j'} the smallest α ln^(l)(n) elements
      from A_j \ ⋃_t B_t which are greater or equal than j';
```

By construction, the sets B_j clearly satisfy all properties required in the statement of the lemma. We thus only need to show that the algorithm cannot fail. In other words, we must prove that the assignment of the elements to B_j from $A_j \setminus \bigcup_t B_t$ is always possible, i.e., that $A_j \setminus \bigcup_t B_t$ always contains enough elements which are eligible. To this aim we will show that the algorithm satisfies the following two invariants for $l = 2,\ldots,l^*$:

- The sets A_j chosen in the first line of the outermost for-loop satisfy:
$$|A_j \setminus \bigcup_t B_t| \ge e^{-(l^*-l)}|A_j|.$$
- For all l: at the end of the outermost for-loop (i.e., after the new sets B_j have been chosen) we have:
$$|T_{l-1,i} \setminus \bigcup_t B_t| \ge e^{-(l^*-l+1/2)}|T_{l-1,i}| \qquad \text{for all } i \ge 1.$$

Technical inequalities
Before we proceed with the actual proof let us state two technical inequalities. As a first auxiliary result we will show that

$$\alpha e^{l^*-l+1} \le \ln^{(l-1)}(n) \qquad \text{for } l < l^*, \tag{9}$$

Let $\delta := l^* - l$. The function $\mathrm{pow}(C,k)$ denotes the k-fold iterated exponentiation defined by $\mathrm{pow}(C,0) = C$ and $\mathrm{pow}(C,k) = e^{\mathrm{pow}(C,k-1)}$ for $k > 0$. By induction on $\delta \ge 1$ it can be proved that

$$1 + \delta + \ln(\alpha) \le \mathrm{pow}(C,\delta-1)$$

if $C \ge 2 + \ln(\alpha)$. Thus, (9) follows by exponentiation of that inequality due to $\mathrm{pow}(C,\delta-1) \le \ln^{(l^*-\delta)}(n) = \ln^{(l)}(n)$.

Obviously it holds that $(\ln(z))^4 \le z$ for z sufficiently large. Hence, we may choose C sufficiently large that

$$(\ln^{(l)}(n))^4 \le \ln^{(l-1)}(n). \tag{10}$$

This is the second auxiliary claim.

Invariants imply existence of B_j
Now we prove that the invariants indeed imply that the sets B_j can always be found. From the first invariant we deduce that

$$|A_j \setminus \bigcup_t B_t| \ge e^{-(l^*-l)}|A_j| \overset{(9)}{\ge} \frac{\alpha(\ln^{(l-1)}(n))^2}{\ln^{(l-1)}(n)} = \alpha\ln^{(l-1)}(n).$$

Hence, for $j \in S_l$ the existence of the desired subset $B_j \subseteq A_j$ follows immediately.

So suppose now that $j \in D_l$ and i denotes the index of the corresponding dense group $T_{l,i} \ni j$. For $j' \in T_{l,i} \cap S_{l+1}$ we may only assign jobs to $B_{j'}$ which are greater or equal than j'. This condition may rule out some jobs in $T_{l,i}$. We consider this by (pessimistically) restricting the eligible jobs to $A_j \setminus (T_{l,i} \cup \bigcup_t B_t)$. Recall that $|T_{l,i}| = (\ln^{(l)}(n))^3$ which is easily checked to be much smaller than $e^{-(l^*-l+1)}|A_j|$ by (9) and (10). It therefore follows from the first invariant that

$$\begin{aligned} |A_j \setminus (T_{l,i} \cup \bigcup_t B_t)| &\ge |A_j \setminus \bigcup_t B_t| - |T_{l,i}| \\ &\ge (1 - 1/e) \cdot e^{-(l^*-l)}|A_j| \\ &\ge e^{-(l^*-l-\ln(1-1/e))}|A_j| \\ &\ge e^{-(l^*-l+1)}|A_j|. \end{aligned}$$

As

$$\begin{aligned} e^{-(l^*-l+1)}|A_j| \overset{(9)}{\ge} \alpha\frac{(\ln^{(l-1)}(n))^2}{\ln^{(l-1)}(n)} &\overset{(10)}{\ge} \alpha[\ln^{(l)}(n)]^4 \\ &= |T_{l,i}| \cdot \alpha\ln^{(l)}(n) \end{aligned}$$

this clearly suffices to find the desired sets B_j for all $j \in T_{l,i} \cap S_{l+1}$.

Proof of invariants

It remains to show that the two invariants hold. Let us consider the first invariant. For $l = l^*$ there is nothing to show, since at that time the sets B_t are still all empty. For $l < l^*$ we deduce the first invariant from the second invariant (applied to $l+1$) in the following way. By definition, $|A_j| = (\ln^{(l-1)}(n))^2$. Hence, we must have $T_{l,i} \subseteq A_j$ for at least $\frac{|A_j|}{|T_{l,i}|} - 2$ different values of i (We prefer to write $|T_{l,i}|$ instead of $(\ln^{(l)}(n))^3$ because $|T_{l,i}|$ depends on i only formally.). Hence,

$$\frac{|A_j|}{|T_{l,i}|} = \frac{(\ln^{(l-1)}(n))^2}{(\ln^{(l)}(n))^3} \overset{(10)}{\geq} \ln^{(l-1)}(n) \geq C. \tag{11}$$

For each $T_{l,i} \subseteq A_j$ we know from the second invariant that $|T_{l,i} \setminus \bigcup_t B_t| \geq e^{-(l^*-l-1/2)}|T_{l,i}|$. Hence, taking the union of the sets $T_{l,i} \subseteq A_j$ we get

$$\begin{aligned} |A_j \setminus \bigcup_t B_t| \quad &\geq \quad e^{-(l^*-l-1/2)}|T_{l,i}| \left(\frac{|A_j|}{|T_{l,i}|} - 2\right) \\ &\overset{(11)}{\geq} \quad e^{-(l^*-l-1/2)}|A_j| \left(1 - \tfrac{2}{C}\right) \geq e^{-(l^*-l)}|A_j| \end{aligned}$$

for C sufficiently large.

Now we show that the second invariant holds as well. First observe that at most

$$\begin{aligned} \max\{\alpha \ln^{(l-1)}(n), |T_{l,i'}| \cdot \alpha \ln^{(l)}(n)\} \quad &= \quad \alpha \ln^{(l-1)}(n) \\ &\overset{(9)}{\leq} \quad e^{-(l^*-l+1)}|A_j| \end{aligned}$$

elements from a set A_j are assigned to a set B_j. Thus, of the $e^{-(l^*-l)}|A_j|$ initially eligible elements of A_j (cf. first invariant)

$$|A_j \setminus \bigcup_t B_t| \geq (1 - 1/e)e^{-(l^*-l)}|A_j| \geq e^{-(l^*-l+0.46)}|A_j|$$

elements stay eligible. At most $\frac{|T_{l-1,i}|}{|A_j|} + 2$ sets A_j have a non-empty intersection with $T_{l-1,i}$. Thus,

$$\frac{|T_{l-1,i}|}{|A_j|} = \frac{(\ln^{(l-1)}(n))^3}{(\ln^{(l-1)}(n))^2} = \ln^{(l-1)}(n) \geq C.$$

Taking C sufficiently large we deduce that

$$\begin{aligned} &|T_{l-1,i} \setminus \bigcup_t B_t| = |T_{l-1,i}| - |T_{l-1,i} \cap \bigcup_t B_t| \\ &\geq \quad |T_{l-1,i}| - \left(\frac{|T_{l-1,i}|}{|A_j|} + 2\right) \cdot (1 - e^{-(l^*-l+0.46)})|A_j| \\ &\geq \quad |T_{l-1,i}| - |T_{l-1,i}| \left(1 + \frac{2}{C}\right) \cdot (1 - e^{-(l^*-l+0.46)}) \\ &\geq \quad e^{-(l^*-l+1/2)}|T_{l-1,i}| \end{aligned}$$

elements in $T_{l-1,i}$ remain eligible after the l-th iteration. □

Now we are in a position to bound the competitive ratio conditional on the fact that the bad events don't occur.

Lemma 7. *There exists a constant ξ_0 such that for all $l = 1, \ldots, L$ the following is true. If the events $\mathcal{A}_1, \ldots, \mathcal{A}_l$ all hold then*

$$\sum_j C_j^{\text{SEPT}} \;\leq\; \xi_0 \cdot \ln^{(l)}(n) \cdot \sum_j C_j^{\text{Opt}}.$$

Proof (Sketch). The main idea is to construct another instance of the scheduling problem in which some of the processing times are reduced. More precisely, we define processing times $p'_1, \ldots, p'_n$ by

$$p'_i := \begin{cases} \nu\mu_i, & \text{if } p_i \geq \nu\mu_i \text{ and } i \leq n - c\ln^{(l)}(n), \\ p_i, & \text{otherwise.} \end{cases}$$

Let $C_j^{\text{Opt}'}$ denote the completion time of job j in an optimum solution of the new problem. It suffices to show that

$$\sum_{j\in J} C_j^{\text{SEPT}} \leq \xi_0 \cdot \ln^{(l)}(n) \cdot \sum_{j\in J} C_j^{\text{Opt}'},$$

for an appropriately chosen constant ξ_0.

Let $w_j^{\text{Opt}'}$ denote the weight of the processing time p'_j in the optimum solution, i.e.,

$$\sum_{j\in J} C_j^{\text{Opt}'} = \sum_{j\in J} w_j^{\text{Opt}'} \cdot p'_j.$$

Note that $w_j^{\text{Opt}'} \geq \frac{n-j+1}{4}$ for $j \leq n - c\ln^{(l)}(n)$ by the definition of $\bar{A}_{l,2}$.

Setting $z := n - c\ln^{(l)}(n)$ we conclude that

$$\begin{aligned} \sum_{j\in J} C_j^{\text{Opt}'} \quad \geq \quad & \underbrace{\sum_{j\in R[z]} \frac{n-j+1}{4} p'_j}_{A} + \underbrace{\sum_{j\in T[z]\setminus S_{\leq l}} \frac{n-j+1}{8} p'_j}_{B} \\ & + \underbrace{\sum_{k=1}^{l} \sum_{j\in S_k} \sum_{i\in B_j} \frac{n-i+1}{8} p'_i}_{C} . + \underbrace{\sum_{j>z} p_j}_{D} . \end{aligned} \tag{12}$$

We now consider these sums separately. The jobs in $R[z]$ are not critical, since $p'_j = p_j$ for $j \in R[z]$. In the sum B we only loose a factor of $\alpha\ln^{(l)}(n)$ since $p_j \leq \alpha\ln^{(l)}(n)$ due to $j \notin S_{\leq l}$. For the third sum we observe that the assumptions on the sets B_j imply that all $j \in S_k$ and $i \in B_j \subseteq T$ satisfy $j \leq i \leq j + \frac{n-j}{5}$ (and hence $\frac{n-i+1}{8} \geq \frac{n-j+1}{10}$) and $\nu\mu_i = p'_i \geq p'_j = \nu\mu_j$. Furthermore, we have that $|B_j| = \alpha\ln^{(k-1)}(n)$ and $p_j < \alpha\ln^{(k-1)}(n)\mu_j$. From that we deduce

$$p'_j \cdot |B_j| = p'_j \cdot \alpha\ln^{(k-1)}(n) = \nu\mu_j\alpha\ln^{(k-1)}(n) > \nu p_j.$$

Using these observations we bound the third sum by $C \geq \sum_{k=1}^{l} \sum_{j\in S_k} \frac{(n-j+1)\nu}{10} p_j$. In sum D we can also lose only a factor of magnitude $c\ln^{(l)}(n)$, since each job $j \geq z$ is counted at least once in an optimal solution and at most $c\ln^{(l)}(n)$ times under SEPT. Combining these bounds the claim follows. □

Now we are finally in a position to prove our main theorem.

Theorem 8. *For the problem $P \mid p_j \sim exp(\lambda_j) \mid \sum C_j$ it holds that*

$$\mathbb{E}\left[\frac{\sum_j C_j^{\text{SEPT}}}{\sum_j C_j^{\text{Opt}}}\right] = \mathcal{O}(1).$$

PROOF. Let us first consider only the single machine case. We define the following partition of our probability space

$$\mathcal{E}_l = \bar{\mathcal{A}}_l \cap \bigcap_{i=1}^{l-1} \mathcal{A}_i \quad \text{for } l = 1, \ldots, L-1 \quad \text{and} \quad \mathcal{E}_L = \bigcap_{i=1}^{L-1} \mathcal{A}_i.$$

As Lemma 7 implies that $(\sum_j C_j^{\text{SEPT}})/(\sum_j C_j^{\text{Opt}})$ is uniformly bounded by $\xi_0 \cdot \ln^{(l-1)}(n)$ whenever $\mathcal{A}_1, \ldots, \mathcal{A}_{l-1}$ hold, we deduce that

$$\begin{aligned} \mathbb{E}\left[\frac{\sum_j C_j^{\text{SEPT}}}{\sum_j C_j^{\text{Opt}}}\right] &= \sum_{l=1}^{L} \mathbb{E}\left[\frac{\sum_j C_j^{\text{SEPT}}}{\sum_j C_j^{\text{Opt}}} \;\middle|\; \mathcal{E}_l\right] \cdot \Pr[\mathcal{E}_l] \\ &= \mathcal{O}\left(\sum_{l=1}^{L-1} \frac{1}{\ln^{(l)}(n)}\right) = \mathcal{O}(1). \qquad (13) \end{aligned}$$

For the parallel machine case with m machines we need an additional lemma (cf. Lemma 9) which shows that, roughly speaking, jobs which are counted w_j-times in the single machine case receive weight w_j/m in the m machine case. □

Since the proof of Theorem 8 relies only on the fact that the distribution of the processing times satisfies $\mathcal{P}(\alpha, \nu)$, we can easily transfer the result to more general distributions. In Section *3.1.4* this is shown for the gamma distribution which comprises the exponential distribution and also the Erlang-distribution as special cases.

3.1.3 Generalization to parallel machines

We prove the parallel machine case of Theorem 8 by modifying the SEPT strategy in the following way: Sort the jobs according to non–decreasing processing times. Assign the i^{th} job to machine ($i \mod m$) as soon as it becomes available. Assume for simplicity's sake that m divides n. Then each machine processes $\frac{n}{m}$ jobs sorted by non–decreasing expected processing time.

Let

$$\begin{aligned} \sum_j C_j^{\text{Opt},m} &= \sum_j w_j^{\text{Opt},m} p_j \qquad \text{and} \\ \sum_j C_j^{\text{SEPT},m} &= \sum_j w_j^{\text{SEPT},m} p_j \end{aligned}$$

denote the objective functions of the SEPT-rule and the optimal offline algorithm in the m-machine case.

LEMMA 9.

$$\frac{\sum_j C_j^{\text{SEPT},m}}{\sum_j C_j^{\text{Opt},m}} \le 2\frac{\sum_j C_j^{\text{SEPT},1}}{\sum_j C_j^{\text{Opt},1}} + 2,$$

PROOF. Recall that the optimal offline solution sorts the jobs by non–decreasing processing time and then places them subsequently on the machine with the least load (cf. [10]). This implies that machine $((k+1) \mod m)$ becomes the machine with the least load and obtains the following job, whenever a job is placed on machine ($k \mod m$).

A job j which is counted $w_j^{\text{Opt},1} =: w_j$ times in the optimal offline solution for one machine is thus counted

$$w'_j := w_j^{\text{Opt},m} = \left\lfloor \tfrac{w_j - 1}{m} \right\rfloor + 1$$

times in the solution for m machines. Hence,

$$\begin{aligned} w'_j &\le \tfrac{w_j-1}{m} + 1 \le \tfrac{w_j}{m} + 1 \qquad \text{and} \\ w'_j &\ge \tfrac{w_j-1}{m} - \left(1 - \tfrac{1}{m}\right) + 1 = \tfrac{w_j}{m}. \end{aligned}$$

These inequalities still hold if we set $w_j = w_j^{\text{SEPT},1}$ and $w'_j = w_j^{\text{SEPT},m}$. We conclude that

$$\begin{aligned} \sum_j C_j^{\text{Opt},m} &= \sum_j w_j^{\text{Opt},m} p_j \\ &\ge \frac{1}{m}\sum_j w_j^{\text{Opt},1} p_j = \frac{1}{m}\sum_j C_j^{\text{Opt},1}. \qquad (14) \end{aligned}$$

For the SEPT rule we deduce that

$$\sum_j C_j^{\text{SEPT},m} \le \frac{1}{m}\sum_j C_j^{\text{SEPT},1} + \sum_j p_j. \qquad (15)$$

Putting (14) and (15) together we obtain that

$$\begin{aligned} \frac{\sum_j C_j^{\text{SEPT},m}}{\sum_j C_j^{\text{Opt},m}} &\le \frac{\frac{1}{m}\sum_j C_j^{\text{SEPT},1} + \sum_j p_j}{\frac{1}{2}\frac{1}{m}\sum_j C_j^{\text{Opt},1} + \frac{1}{2}\sum_j C_j^{\text{Opt},m}} \\ &\le 2\frac{\sum_j C_j^{\text{SEPT},1}}{\sum_j C_j^{\text{Opt},1}} + 2\frac{\sum_j p_j}{\sum_j C_j^{\text{Opt},m}} \\ &\le 2\frac{\sum_j C_j^{\text{SEPT},1}}{\sum_j C_j^{\text{Opt},1}} + 2, \end{aligned}$$

which completes the proof of the lemma. □

From Lemma 9 the parallel machine case of Theorem 8 follows immediately.

3.1.4 Generalization to gamma-distributed processing times

Since the proof of Theorem 8 relies merely on the fact that $\mathcal{P}(\alpha, \nu)$ holds for the distribution of the processing times we can easily transfer the result to more general distributions.

Erlang distributions are often chosen to model job execution times, e.g. in queuing theory, and they thus represent a natural candidate for such a generalization. The following lemma shows that $\mathcal{P}(\alpha, \nu)$ is satisfied by an even broader class of distributions, which contains for example all Erlang distributions but also much more, namely for all gamma distributions (with certain bounds on the parameters).

LEMMA 10. *For $\alpha_{\min}, \alpha_{\max} > 0$ there are $\alpha, \nu > 0$ such that all gamma distributions with parameters $a, b > 0$ and $\alpha_{\min} \le a \le \alpha_{\max}$ satisfy $\mathcal{P}(\alpha, \nu)$.*

PROOF. The gamma distribution with parameters $a, b > 0$ has the density function

$$f(x \mid a, b) = \frac{1}{b^a \Gamma(a)} \cdot e^{-x/b} \cdot x^{a-1},$$

where $\Gamma(a) = \int_0^\infty x^{a-1} \cdot e^{-x}\, \mathrm{d}x$ denotes the gamma function. The incomplete gamma function $\Gamma(a, z)$ is defined by

$$\Gamma(a, z) = \int_z^\infty x^{a-1} \cdot e^{-x}\, \mathrm{d}x.$$

We use the following approximation for $\Gamma(a, z)$ from [1] (see 6.5.32):

$$\Gamma(a, z) = z^{a-1} \cdot e^{-z} \cdot (1 + \mathcal{O}(1/z)).$$

For z sufficiently large we obtain that

$$\Gamma(a, z) \le e^{-z/2}. \qquad (16)$$

Let X be a gamma distributed random variable. Note that $\mu := \mathbb{E}[X] = ab$. We deduce

$$\begin{aligned} \Pr[X \geq z] &= \int_z^\infty f(x \mid a, b)\,\mathrm{d}x \\ &= \frac{1}{b^a \Gamma(a)} \cdot \int_z^\infty e^{-x/b} \cdot x^{a-1}\,\mathrm{d}x \\ &= \frac{1}{\Gamma(a)} \cdot \Gamma(a, z/b) \end{aligned}$$

Therefore,

$$\Pr[X \geq \alpha \ln(z)\mu] = \frac{1}{\Gamma(a)} \cdot \Gamma(a, \alpha \ln(z) a)$$

Using (16) it follows that

$$\Pr[X \geq \alpha \ln(z)\mu] \leq \frac{1}{\Gamma(a)} \cdot e^{-\alpha \ln(z) a/2} \leq z^{-5}$$

for $\alpha \geq 10/a$ and z sufficiently large. Note that

$$\Pr[X \geq \nu\mu] = \Pr[X \geq \nu ab] = \frac{\Gamma(a, \nu a)}{\Gamma(a)} =: g(a),$$

and $\lim_{\nu \to 0} g(a) = 1$. Hence, for sufficiently small ν we obtain that $\Pr[X \geq \nu\mu] \geq 1/2$. Since a is bounded by $\alpha_{\min}$ and $\alpha_{\max}$, we can find α and μ such that $\mathcal{P}(\alpha, \nu)$ is satisfied for all parameters a, b in the allowed range. $\square$

Note that the bounds on the parameter a in Lemma 10 still admit arbitrarily large expectation and variance, since $\mathbb{E}[X] = ab$ and $\mathrm{Var}[X] = ab^2$, and there are no restrictions on the choice for b.

4. REFERENCES

[1] M. Abramowitz and I. A. Stegun, editors. *Handbook of Mathematical Functions*. Dover Publications, New York, 1970.

[2] E. G. Coffmann, Jr. and E. N. Gilbert. Expected relative performance of list scheduling rules. *Oper. Res.*, 33:548–561, 1985.

[3] E. G. Coffmann, Jr., K. Sao, M. Hofri, and A. C. Yao. A stochastic model of bin–packing. *Inf. and Cont.*, 44:105–115, 1980.

[4] W. Feller. *An Introduction to Probability Theory and its Applications*, volume I. John Wiley & Sons, New York, 1957.

[5] K. D. Glazebrook. Scheduling tasks with exponential service times on parallel machines. *Journal of Applied Probability*, 16:685–689, 1979.

[6] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnoy Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5:287–326, 1979.

[7] T. Kämpe. On the optimality of static priority policies in stochastic scheduling on parallel machines. *Journal of Applied Probability*, 24:430–448, 1987.

[8] N. Karmakar. Probabilistic analysis of some bin–packing algorithms. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pages 107–111, 1982.

[9] Z. Liu and E. Sanlaville. Stochastic scheduling with variable profile and precedence constraints. Technical Report RR-1525, Inria, Institut National de Recherche en Informatique et en Automatique, 1997.

[10] R. McNaughton. Scheduling with deadlines and loss functions. *Management Science*, 6:1–12, 1959.

[11] R. H. Möhring and F. J. Radermacher. Introduction to stochastic scheduling problems. In *Contributions to operations research*, volume 240 of *Lect. Notes Econ. Math. Syst.*, pages 72–130, Oberwolfach/Ger., 1985.

[12] R. H. Möhring, F. J. Radermacher, and G. Weiss. Stochastic scheduling problems I: General strategies. *Z. Oper. Res.*, Ser. A. 28:193–260, 1984.

[13] R. H. Möhring, F. J. Radermacher, and G. Weiss. Stochastic scheduling problems II: General strategies. *Z. Oper. Res.*, Ser. A. 29:65–104, 1984.

[14] R. H. Möhring, A. Schulz, and M. Uetz. Approximation in stochastic scheduling: The power of LP–based priority policies. *Journal of the ACM*, 46:924–942, 1999.

[15] C. H. Papadimitriou and J. N. Tsitsiklis. On stochastic scheduling with in–tree precendence constraints. *SIAM Journal on Computing*, 16:1–6, 1987.

[16] M. H. Rothkopf. Scheduling with random service times. *Management Science*, 12:703–713, 1966.

[17] W. E. Smith. Various optimizers for single–stage production. *Naval Research and Logistics Quaterly*, 3:59–66, 1956.

[18] R. R. Weber, P. Varaiya, and J. Walrand. Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flowtime. *Journal of Applied Probability*, 23:841–847, 1986.

[19] G. Weiss and M. Pinedo. Scheduling tasks with exponential service times on non–identical processors to minimize various cost functions. *Journal of Applied Probability*, 17:187–202, 1980.