

Lower bounds for Smith’s rule in stochastic machine scheduling

Caroline Jagtenberg¹, Uwe Schwiegelshohn², and Marc Uetz³

¹ Utrecht University, Dept. of Mathematics, P.O. Box 80010, 3508 TA Utrecht, The Netherlands, carolinetjes@gmail.com

² TU Dortmund, Robotics Research Institute, 44227 Dortmund, Germany, uwe.schwiegelshohn@udo.edu

³ University of Twente, Dept. of Applied Mathematics, P.O. Box 217, 7500 AE Enschede, The Netherlands, m.uetz@utwente.nl

Abstract We consider the problem to minimize the weighted sum of completion times in nonpreemptive parallel machine scheduling. In a landmark paper from 1986, Kawaguchi and Kyan [5] showed that scheduling the jobs according to the WSPT rule –also known as Smith’s rule– has a performance guarantee of $\frac{1}{2}(1 + \sqrt{2}) \approx 1.207$. They also gave an instance to show that this bound is tight. We consider the stochastic variant of this problem in which the processing times are exponentially distributed random variables. We show, somehow counterintuitively, that the performance guarantee of the WSEPT rule, the stochastic analogue of WSPT, is not better than 1.229. This constitutes the first lower bound for WSEPT in this setting, and in particular, it shows that even with exponentially distributed processing times, stochastic scheduling has somewhat nastier worst-case examples than deterministic scheduling. In that respect, our analysis sheds new light on the fundamental differences between deterministic and stochastic scheduling.

Keywords. *stochastic scheduling, WSEPT, exponential distribution*

1 Introduction

Minimizing the weighted sum of completion times on m parallel, identical machines is an archetypical problem in the theory of scheduling. In this problem, we are given n jobs which have to be processed non-preemptively on m machines. Each job j comes with a processing time p_j and a weight w_j , and when C_j denotes job j ’s completion time in a given schedule, the goal is to compute a schedule that minimizes the total weighted completion time $\sum_j w_j C_j$. In the classical 3-field notation for scheduling problems [3], the problem is denoted $P \parallel \sum w_j C_j$. For a single machine, a simple exchange argument shows that scheduling the jobs in order of nonincreasing ratios w_j/p_j gives the optimal schedule [11]. Greedily scheduling the jobs in this order is known as WSPT rule or Smith’s rule. On parallel identical machines, WSPT is known to be a $\frac{1}{2}(1 + \sqrt{2})$ -approximation, and this bound is tight [5]. The computational tractability of the problem was

finally settled by showing the existence of a PTAS [10], given that the problem is strongly NP-complete if m is part of the input [2].

In this paper, we consider the stochastic variant of the problem. It is assumed that the processing time p_j of a job j is not known in advance. It becomes known upon completion of the job. Only the distribution of the corresponding random variable P_j , or at least its expectation $E[P_j]$, is given beforehand. More specifically, we assume that the processing times of jobs are governed by independent, exponentially distributed random variables. That is to say, each job comes with a parameter $\lambda_j > 0$, and

$$P[P_j > t] = e^{-\lambda_j t}.$$

We denote that by writing $P_j \sim \exp(\lambda_j)$. Exponentially distributed processing times somehow represent the cream of stochastic scheduling, in particular when juxtaposing stochastic and deterministic scheduling: The exponential distribution is characterized by the memory-less property, that is,

$$P[P_j > s + t | P_j > s] = P[P_j > t].$$

So for any non-finished job it is irrelevant how much processing it has already received. This is obviously a decisive difference to deterministic scheduling models, and puts stochastic scheduling apart. Next to that, the model with exponentially distributed processing times is attractive because it makes the stochastic model analytically tractable.

In the stochastic setting, the analogue of Smith's rule is greedily scheduling the jobs in order of non-increasing ratios $w_j/E[P_j]$, also called WSEPT [8]. For a single machine, this is again optimal [9]. For parallel machines, it has been shown that the WSEPT rule achieves a performance bound of $(2 - \frac{1}{m})$ within the class of all non-anticipatory stochastic scheduling policies [7]. That is to say, if Π^* denotes an optimal non-anticipatory stochastic scheduling policy, then

$$E[\sum w_j C_j^{\text{WSEPT}}] \leq \left(2 - \frac{1}{m}\right) E[\sum w_j C_j^{\Pi^*}].$$

We refer e.g. to [6] for precise definitions on non-anticipatory stochastic scheduling policies. For the purpose of this paper, it suffices to know that non-anticipatory stochastic scheduling policies are online in the sense that they are, at any given time t , only allowed to use information that is available at that time t . Obviously, this is also the case for WSEPT, as the distributions P_j , and particularly $E[P_j]$ are available beforehand.

The major purpose of this paper is to establish the first lower bound for the $(2 - \frac{1}{m})$ performance guarantee of [7] for exponentially distributed processing times. In fact, we are not aware of any result in this direction. The only result known to us is an instance showing that WSEPT can miss the optimum by a factor 3/2, but then for arbitrary processing time distributions [12, Ex. 3.5.12]. We show that there are instances where WSEPT misses the optimum by at least a factor 1.229. To obtain our result, we carefully adapt and analyze the worst-case instance of [5]. Note that the originality of this result lies in the fact

that $1.229 > \frac{1}{2}(1 + \sqrt{2}) \approx 1.207$. Hence, stochastic scheduling with exponentially distributed processing times has worse worst-case instances than deterministic scheduling.

This observation may seem somewhat counterintuitive: Observe that for unit weights where $w_j = 1$, the SPT rule is optimal for minimizing $\sum_j C_j$ in the deterministic setting [8], and also SEPT is optimal for minimizing $E[\sum_j C_j]$ when processing times are exponentially distributed [1]. For exponentially distributed processing times and weights that are agreeable in the sense that there exists an ordering such that $w_1 \geq \dots \geq w_n$ and $w_1 \lambda_1 \geq \dots \geq w_n \lambda_n$, scheduling the jobs in this WSEPT order is optimal [4], while the corresponding deterministic problem is already NP-hard, and in particular, WSPT is not optimal. That is to say, there are examples where the stochastic version with exponentially distributed processing times is computationally easier than the deterministic version of the same problem. Our result shows that with arbitrary weights, the situation is again fundamentally different. Next to this qualitatively new insight, our analysis also sheds light on phenomena in stochastic scheduling which are interesting in their own right.

The paper is organized as follows. In Section 2, we briefly review the worst-case instance presented in [5]. We derive several technical lemmas about scheduling jobs with exponentially distributed processing times in Section 3. Section 4 presents the analysis of the stochastified instance of [5], and finally, Section 5 summarizes our conclusions.

2 The Kawaguchi & Kyan instance

We briefly summarize the instance from [5] that achieves the bound $\frac{1}{2}(1 + \sqrt{2})$ for deterministic scheduling, as our stochastic instance is a stochastic variant thereof. Let n be the number of jobs and m the number of machines. Denote the processing time of job j by p_j and its weight by w_j . The (deterministic) instance is then given by:

$$\begin{aligned} m &= m^* + \lfloor (1 + \sqrt{2})m^* \rfloor \\ n &= mn^* + m^* \\ p_j &= w_j = 1/n^* && \text{for } 1 \leq j \leq mn^* \\ p_j &= w_j = 1 + \sqrt{2} && \text{for } mn^* + 1 \leq j \leq mn^* + m^* \end{aligned}$$

Here, m^* denotes an integer, and n^* is an integer that can be divided by $\lfloor (1 + \sqrt{2})m^* \rfloor$. Notice that $\frac{w_j}{p_j} = 1$ for all j . This means that any list schedule is in fact a WSPT schedule. Let us refer to the first mn^* jobs as short jobs, and the remaining m^* jobs as long jobs.

Let M_L be the total weighted completion time of a schedule in which all short jobs are processed first, and M^* be the total weighted completion time of a schedule where the long jobs are processed first. Figure 1 depicts these two schedules. The schedule on the left of Figure 1 has value M^* . Here the last jobs of length $1/n^*$ finishes at time $t \approx 1.4$ (for large values of m and n^*).

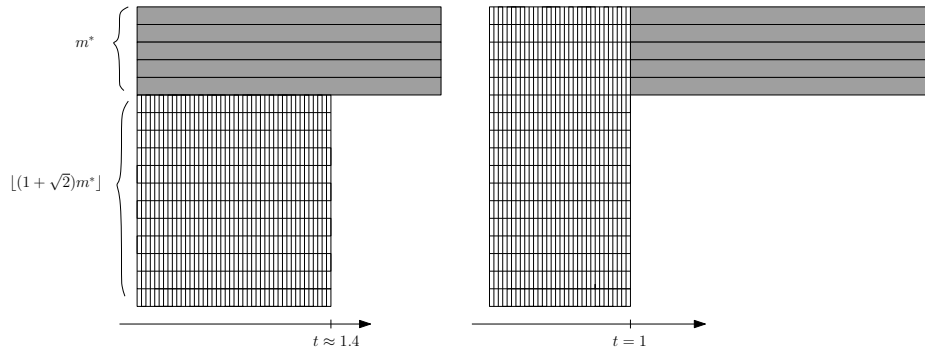


Figure 1. Two different WSPT schedules with values M^* and M_L respectively.

The schedule on the right of Figure 1 has value M_L , it finishes the last jobs of length $1/n^*$ exactly at time 1. In Figure 1 we used $m^* = 5$ and $n^* = 32$. It can be verified (see [5]) that $M_L = (1 + \sqrt{2})(2 + \sqrt{2})m^* + (m/2)(1 + 1/n^*)$ and $M^* = (1 + \sqrt{2})^2 m^* + (m/2)(m/\lfloor(1 + \sqrt{2})m^*\rfloor + 1/n^*)$. The ratio M_L/M^* then tends to $(1 + \sqrt{2})/2$ as $m^* \rightarrow \infty$ and $n^* \rightarrow \infty$.

3 Preliminaries on jobs with exponentially distributed processing times

The crucial insight when stochastifying the instance by Kawaguchi and Kyan is the following. The schedule with value M_L is essentially identical to the expected situation in stochastic scheduling. However, the schedule with value M^* has a significantly different realization with exponentially distributed processing times. This is expressed in the following lemmas, where λ and x are arbitrary positive parameters. In the following we denote by

$$H_n := \sum_{i=1}^n \frac{1}{i}$$

the n th harmonic number, with $H_0 := 0$.

The first lemma gives an estimate on expected job completion times for parallel jobs with $P_j \sim \exp(\lambda)$.

Lemma 1. *When scheduling in parallel m jobs with i.i.d. exponential processing times $P_j \sim \exp(\lambda)$, the expected number $m(t)$ of machines that are idle at a given time t is bounded as follows,*

$$m(t) \geq \lfloor m(1 - e^{-\lambda t}) \rfloor.$$

Proof. The first completion time is distributed as the minimum of m independent $\exp(\lambda)$ distributions. This is an $\exp(m\lambda)$ distribution, hence it is expected at time $t_1 = \frac{1}{m\lambda}$. After the first job completion, we have $m - 1$ jobs running. Since the

exponential distribution is memoryless, the next completion is expected a time $\frac{1}{(m-1)\lambda}$ later, so $t_2 = \frac{1}{m\lambda} + \frac{1}{(m-1)\lambda}$. By continuing this argument we find that the k th job completion is expected at time

$$t_k = \sum_{j=1}^k \frac{1}{(m-j+1)\lambda} = \frac{1}{\lambda} \sum_{l=m-k+1}^m \frac{1}{l} = \frac{1}{\lambda} (H_m - H_{m-k}). \quad (1)$$

Note that $m(t_k) = k$, for $k = 1, \dots, m$. We now use that $H_i \geq \ln(i) + \gamma$ for all i , where

$$\gamma := \lim_{i \rightarrow \infty} (H_i - \ln i) \approx 0.57721$$

denotes the Euler-Mascheroni constant. Furthermore, $H_i - \ln(i)$ is monotonically decreasing in i . Hence we may conclude that

$$t_k \leq \frac{1}{\lambda} (\ln(m) + \gamma - \ln(m-k) - \gamma) = \frac{1}{\lambda} \ln \left(\frac{m}{m-k} \right). \quad (2)$$

Here, k is the expected number of finished jobs at time t_k . Hence, (2) yields

$$m(t_k) = k \geq m(1 - e^{-\lambda t_k}) \quad (3)$$

for $k = 1, 2, \dots, m$. Together with the fact that $m(t)$ is integer valued, (3) yields

$$m(t) \geq \lfloor m(1 - e^{-t\lambda}) \rfloor.$$

for all $t \geq 0$. □

Note that the last job is expected to finish at time $\frac{1}{\lambda} \Theta(\log m)$. Nevertheless, the average expected completion time of the jobs is $1/\lambda$; see also Figure 2 for an illustration.

Lemma 2. *Consider n^*T jobs with i.i.d. processing times $P_j \sim \exp(n^*)$ and weights $w_j = 1/n^*$, scheduled on a single machine. Then for all $\varepsilon > 0$ there exists n^* large enough so that*

$$\mathbb{E} \left[\sum_j w_j C_j \right] \leq \int_0^T t \, dt + \varepsilon.$$

Proof. As there is an expected job completion each $1/n^*$ time steps, one easily calculates that $\mathbb{E} \left[\sum_j w_j C_j \right] = \frac{1}{2}T^2 + \frac{1}{2n^*}T$, so for $n^* \geq \frac{T}{2\varepsilon}$ the claim is true. □

Lemma 3. *Let $m(t) \geq 0$ denote the number of available machines at time t , and assume $m(t)$ is non-decreasing. When greedily scheduling jobs with i.i.d. processing times $P_j \sim \exp(n^*)$ and weights $w_j = 1/n^*$ from time 0 until T on the available machines, for all $\varepsilon > 0$ there exists n^* large enough so that*

$$\mathbb{E} \left[\sum_j w_j C_j \right] \leq \int_0^T m(t) t \, dt + \varepsilon.$$

Proof. Let T_i ($i = 0, 1, 2, \dots$) be the times that a new machine becomes available, with $T_0 := 0$. For n^* large enough, we expect $m(T_i)n^*(T_{i+1} - T_i)$ jobs to be scheduled between times T_i and T_{i+1} . It is straightforward to extend Lemma 2 to this case, which yields

$$\mathbb{E} \left[\sum_j w_j C_j \right] \leq m(T_i) \int_{T_i}^{T_{i+1}} t \, dt + \varepsilon_i.$$

Therefore we get

$$\mathbb{E} \left[\sum_j w_j C_j \right] \leq \sum_i m(T_i) \int_{T_i}^{T_{i+1}} t \, dt + \varepsilon_i = \int_0^T m(t) t \, dt + \sum_i \varepsilon_i.$$

So for $\varepsilon = \sum_i \varepsilon_i$ and n^* accordingly large, the claim is true. \square

The next lemma is concerned with the total weighted completion time of short jobs that succeed a set of long jobs.

Lemma 4. *Suppose we first schedule m i.i.d. long jobs with processing times $P_j \sim \exp(\lambda)$, followed by $xm n^*$ i.i.d. short jobs, with processing times $P_j \sim \exp(n^*)$ and weights $w_j = 1/n^*$, where n^* is large. Let \mathcal{X} be the expected weighted sum of completion times of the short jobs. Then for any T such that $\frac{1}{\lambda}(e^{-\lambda T} - 1) + \frac{(m-1)T}{m} \geq x$, and n^* large enough we have that*

$$\mathcal{X} \leq \int_0^T (m(1 - e^{-\lambda t}) - 1) t \, dt. \quad (4)$$

Proof. Denote by $m(t)$ the number of machines at time t that are available for processing short jobs, and by T^* the earliest point in time such that we can expect all short jobs to be finished by time T^* . Notice that the total expected processing of short jobs equals xm . Therefore, for n^* large enough, T^* can be approximated arbitrarily well by the solution of the equation

$$xm = \int_0^{T^*} m(t) \, dt. \quad (5)$$

With T^* as in (5), Lemma 3 yields that $\mathcal{X} \leq \int_0^{T^*} m(t) t \, dt + \varepsilon$. Now recall that $m(t)$ can be bounded as in Lemma 1. Define

$$f(t) = m(1 - e^{-\lambda t}) - 1. \quad (6)$$

Then $m(t) \geq f(t)$ for all $t \geq 0$. If we require for T that

$$\begin{aligned} \int_0^T f(t) \, dt &\geq xm \\ \Leftrightarrow \frac{1}{\lambda}(e^{-\lambda T} - 1) + \frac{(m-1)T}{m} &\geq x, \end{aligned}$$

then $xm = \int_0^{T^*} m(t) dt \leq \int_0^T f(t) dt$. We therefore conclude that

$$\int_0^{T^*} m(t) t dt < \int_0^T f(t) t dt, \quad (7)$$

because $m(t) \geq f(t)$, and $m(t)$ is a step function while $f(t)$ is continuous. Intuitively, the expression $\int_0^T f(t) t dt$ equals the total weighted sum of completion times for infinitesimally small jobs (i.e., when $n^* \rightarrow \infty$), with total expected processing at least xm , scheduled on a set of “machines” that become available no earlier than $m(t)$. We finally conclude from (7) that

$$\mathcal{X} \leq \int_0^{T^*} m(t) t dt + \varepsilon \leq \int_0^T f(t) t dt,$$

because ε can be chosen arbitrarily small for n^* large enough. □

A variation of this lemma will be used later in the analysis. Notice that the technical condition on T as stated in Lemma 4 only makes sure that all short jobs can be processed by time T when the machine availability is governed by $f(t)$ rather than $m(t)$. The same approach will be used also in Section 4.

Finally, we make a statement about scheduling a block of (short) jobs.

Lemma 5. *Suppose we list schedule $xm n^*$ i.i.d. short jobs with processing times $P_j \sim \exp(n^*)$ greedily on m machines. Then the average expected machine completion time equals x , and for any $\delta > 0$ there exists n^* large enough such that the earliest expected machine completion time is at time $t \geq x - \delta$.*

Proof. The claim about the average expected machine completion time is clear, because the total expected processing is xm . For the second claim, consider the first time, say t , that a machine runs out of jobs. Then there are $m - 1$ jobs still in process. We know from Lemma 1 that the last machine that runs out of jobs is expected to be at time $t + \sum_{i=1}^{m-1} \frac{1}{i n^*}$. For m large enough, we have $\sum_{i=1}^{m-1} \frac{1}{i n^*} \leq \frac{1}{n^*} [\ln(m) + \gamma]$. So for $n^* \geq (m-1)/(\delta(\ln(m) + \gamma))$, the last machine completion time is expected no later than $t + \delta/(m-1)$. Now the claim follows, as the average machine completion time is x . □

4 The stochastic instance

Even though other instances may lead to comparable results, we find it instructive to consider precisely the stochastic analogue of the instance presented by Kawaguchi and Kyan [5]. Indeed, it turns out that the analyses for such instances use identical arguments, the core of which is represented by the lemmas given in Section 3.

We keep all parameters the same as in Section 2, except that the processing times of long jobs will be $P_j \sim \exp(1/(1 + \sqrt{2}))$, and the processing times of short jobs will be $P_j \sim \exp(n^*)$. So the expected processing times of long and short jobs are identical to the deterministic processing times in [5].

4.1 Intuition about the schedules

Suppose we start all long jobs first and then fill up the remaining machines with short jobs. By Lemma 1 we expect the i^{th} long job to finish at time:

$$t_i = \sum_{j=1}^i \frac{1 + \sqrt{2}}{m^* - j + 1} \quad (8)$$

Therefore, we expect the last short job to be completed significantly earlier than in the deterministic case. For a finite number of machines, the schedule will look like depicted in Figure 2. The crucial point is that the average expected time that

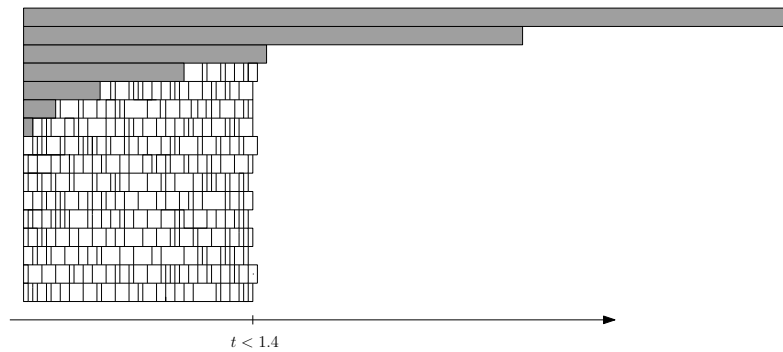


Figure 2. Schedule with all long jobs starting at time 0.

machines finish processing short jobs will be smaller than in the deterministic case. This happens because many long jobs finish much earlier, and the late finishing of few long jobs doesn't matter for the short jobs. Hence, the contribution of the short jobs will decrease when compared to the deterministic case.

Suppose on the other hand that we first start all the short jobs. The set of short jobs is not likely to produce the ideal rectangle as it did in the deterministic case. However, as suggested by Lemma 5 the gap between the time the first machine runs out of short jobs and the time the last machine runs out of short jobs can be bounded. And because we can freely choose n^* , the inverse of the expected processing time of short jobs, the expected deviation from the ideal rectangle can be made negligible by letting n^* be large enough. See Figure 3 for an example. The crucial point is that, in this situation, the expected cost of the schedule is almost equivalent to the deterministic case.

4.2 Lower bound on performance of WSEPT

Let S^* denote the objective value $E \left[\sum_j w_j C_j \right] = \sum_j w_j E[C_j]$ for the case when we first schedule all long jobs. Similarly, let S_L denote the objective value

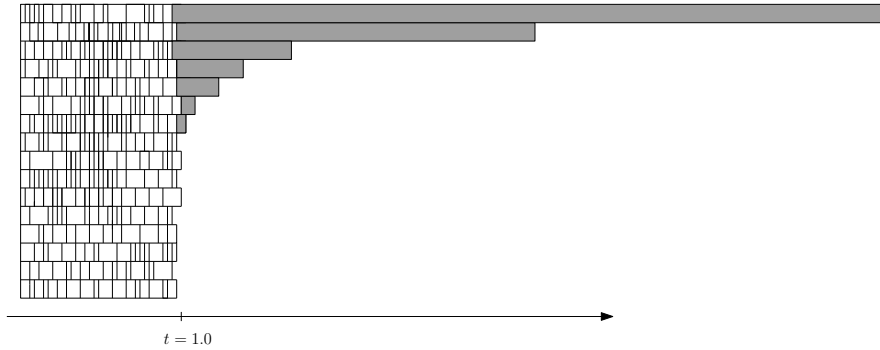


Figure 3. Schedule with all long jobs starting only after short jobs.

for the schedule that starts long jobs only when there is no short job left to be scheduled. S^* is in fact optimal, whereas S_L is the worst case, but this is inessential. Both are in fact WSEPT, hence the ratio S_L/S^* is a lower bound for the approximation ratio of the WSEPT rule in stochastic machine scheduling with exponentially distributed processing times.

We choose m^* sufficiently large, and n^* , a multiple of $\lfloor (1 + \sqrt{2})m^* \rfloor$, we may choose arbitrarily large in comparison to m^* (i.e., $n^* \gg m^*$). In fact, we can make the choice of these two parameters in such a way that all our technical lemmas from Section 3 do apply.

The optimal case, S^* . We split S^* up into the contribution of the long jobs S_{long}^* and the contribution of the short jobs S_{short}^* . So

$$S^* = S_{long}^* + S_{short}^* \quad (9)$$

The value of S_{long}^ .* We start all m^* long jobs at time 0. Their expected completion time is $1 + \sqrt{2}$ each. Hence the contribution of the long jobs is simply given by

$$S_{long}^* = m^*(1 + \sqrt{2})^2, \quad (10)$$

which is actually the same as in the deterministic case.

The value of S_{short}^ .* This is a bit more complicated to calculate. We expect the short jobs to be located in the red and blue area, as depicted in Figure 4.

The expected total processing of short jobs \mathcal{B} that fit in the blue rectangle is given by

$$\mathcal{B} = \int_0^T (m - m^*) dt$$

According to (6) in the proof of Lemma 4, the number of finished long jobs at time $t \geq 0$ is at least:

$$f(t) = m^*(1 - e^{-t/(1+\sqrt{2})}) - 1.$$

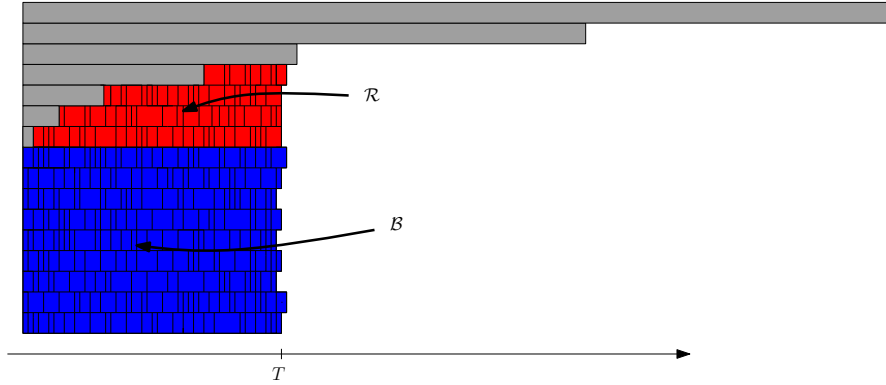


Figure 4. For T large enough, all short jobs fit in the red and blue areas \mathcal{R} and \mathcal{B} .

Therefore, the expected total processing of short jobs \mathcal{R} that fit in the red area is bounded by

$$\mathcal{R} \geq \int_0^T f(t) dt$$

We want to find a value for T such that all short jobs are expected to be finished by T , i.e. $\mathcal{B} + \mathcal{R} \geq m$. We have not attempted to solve this equation analytically, but one can easily check that

$$T = 1.2933 \tag{11}$$

suffices.

Then S_{short}^* , the expected weighted sum of completion times for all mn^* short jobs, is similar to \mathcal{X} in Lemma 4. We now find, for m^* and n^* sufficiently large,

$$S_{short}^* \leq \int_0^T (m - m^*) t dt + \int_0^T f(t) t dt. \tag{12}$$

With (11) and (12) we can calculate

$$S_{short}^* \leq 2.266m^* - 0.836. \tag{13}$$

Combining (10) and (13) gives

$$S^* = S_{long}^* + S_{short}^* \leq (1 + \sqrt{2})^2 m^* + 2.266m^* - 0.836. \tag{14}$$

The worst case, S_L . Now we switch to the case where we first schedule all the short jobs. Again split the objective value into the two parts contributed by the short and long jobs, respectively.

$$S_L = S_L^{short} + S_L^{long}.$$

The value of S_L^{short} : We have m machines working on mn^* jobs with processing times $P_j \sim \exp(n^*)$. According to Lemma 5, on average a machine is expected to finish with these jobs at time 1, and for any $\delta > 0$, we can find n^* large enough so that we expect all machines to be filled with short jobs at least until time $1 - \delta$. Hence, we conclude that the average expected completion time of a short job is arbitrarily close to $1/2$. Therefore, for any $\varepsilon > 0$, there is n^* large enough so that

$$S_L^{short} \geq \frac{m}{2} - \varepsilon/2. \quad (15)$$

The value of S_L^{long} : Remember that the schedule is expected to look like depicted in Figure 3. Using Lemma 5 again, we know that long jobs are expected to start no earlier than $1 - \delta$. So by assuming they all start at this time, we get a lower bound for their completion times (and also for S_L^{long}). If all long jobs start at $1 - \delta$, the average expected completion time is $2 - \delta + \sqrt{2}$. Multiplying this by the weight and summing over all m^* jobs, we may conclude that for any $\varepsilon > 0$ there is n^* large enough so that

$$S_L^{long} \geq (2 + \sqrt{2})(1 + \sqrt{2})m^* - \varepsilon/2. \quad (16)$$

With (15) and (16) we now have

$$S_L = S_L^{short} + S_L^{long} \geq \frac{m}{2} + (2 + \sqrt{2})(1 + \sqrt{2})m^* - \varepsilon. \quad (17)$$

The ratio. Finally, let α be the approximation ratio of Smith's rule for exponentially distributed processing times. Then

$$\alpha \geq \frac{S_L}{S^*}.$$

Remember that $m = m^* + \lfloor (1 + \sqrt{2})m^* \rfloor$. Now for carefully chosen $n^* \gg m^*$, and taking $m^* \rightarrow \infty$, equations (14) and (17) give

$$\frac{S_L}{S^*} \geq \frac{m/2 + (2 + \sqrt{2})(1 + \sqrt{2})m^* - \varepsilon}{(1 + \sqrt{2})^2 m^* + 2.266m^* - 0.836} > 1.229.$$

So we conclude that $\alpha > 1.229$. Note that this is strictly larger than the approximation ratio for WSPT in the deterministic case, which is 1.207.

5 Conclusion

For the purpose of this paper, we opted to focus on the qualitative result that WSEPT performs worse than WSPT, and found it instructive to look at the stochasticification of the classical instance by Kawaguchi and Kyan.

Interestingly, we also found instances (not discussed in this paper) —with comparable building blocks and features— where WSPT is always optimal for

the deterministic case, while WSEPT is not optimal for the stochastic counterpart with exponentially distributed processing times.

In conclusion, small improvements in the ratio 1.229 may be possible, given that the parameters we use are optimized for the deterministic setting. This will be part of the full version of this paper. Yet, our first impression after playing with several variations of the instance is that the upper bound $(2 - 1/m)$ seems out of reach.

This naturally leads to the question to improve the analysis of WSEPT in stochastic machine scheduling. In that respect, it is interesting to note that the analysis of [7] does not specifically exploit the exponential distribution; the result is also valid for more general distributions.

Acknowledgements

The second and third author thank the organizers of the 2010 Dagstuhl Seminar on Scheduling.

References

1. J. L. Bruno, P. J. Downey, and G. N. Frederickson. Sequencing tasks with exponential service times to minimize the expected flowtime or makespan. *Journal of the Association for Computing Machinery*, 28:100–113, 1981.
2. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, 1979.
3. R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5:287–326, 1979.
4. T. Kämpke. On the optimality of static priority policies in stochastic scheduling on parallel machines. *Journal of Applied Probability*, 24:430–448, 1987.
5. T. Kawaguchi and S. Kyan. Worst case bound on an LRF schedule for the mean weighted flow-time problem. *SIAM Journal on Computing*, 15:1119–1129, 1986.
6. R. H. Möhring, F. J. Radermacher, and G. Weiss. Stochastic scheduling problems I: General strategies. *ZOR - Zeitschrift für Operations Research*, 28:193–260, 1984.
7. R. H. Möhring, A. S. Schulz, and M. Uetz. Approximation in stochastic scheduling: The power of LP-based priority policies. *Journal of the Association for Computing Machinery*, 46:924–942, 1999.
8. M. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Prentice-Hall, Upper Saddle River (NJ), 2 edition, 2002.
9. M. H. Rothkopf. Scheduling with random service times. *Management Science*, 12:703–713, 1966.
10. M. Skutella and G. J. Woeginger. A PTAS for minimizing the total weighted completion time on identical parallel machines. *Mathematics of Operations Research*, 25:63–75, 2000.
11. W. E. Smith. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3:59–66, 1956.
12. M. Uetz. *Algorithms for Deterministic and Stochastic Scheduling*. PhD thesis, Institut für Mathematik, Technische Universität Berlin, Berlin, Germany, 2001. Published: Cuvillier Verlag, Göttingen, Germany.