Operations Research Letters 41 (2013) 570–575

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

Analysis of Smith's rule in stochastic machine scheduling

Caroline Jagtenberg[a], Uwe Schwiegelshohn[b], Marc Uetz[c,*]

[a] CWI, Science Park 123, 1098 XG Amsterdam, The Netherlands
[b] TU Dortmund, Robotics Research Institute, 44227 Dortmund, Germany
[c] University of Twente, Applied Mathematics, P.O. Box 217, 7500 AE Enschede, The Netherlands

ARTICLE INFO

Article history:
Received 18 February 2013
Received in revised form
30 July 2013
Accepted 2 August 2013
Available online 9 August 2013

Keywords:
Stochastic scheduling
WSEPT
Exponential distribution

ABSTRACT

In a landmark paper from 1986, Kawaguchi and Kyan show that scheduling jobs according to ratios weight over processing time – also known as Smith's rule – has a tight performance guarantee of $(1+\sqrt{2})/2 \approx 1.207$ for minimizing the weighted sum of completion times in parallel machine scheduling. We prove the counterintuitive result that the performance guarantee of Smith's rule is not better than 1.243 when processing times are exponentially distributed.

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 $\sum_j w_jC_j$. In the classical 3-field notation for scheduling problems [5], the problem is denoted by $P||\sum w_jC_j$. For a single machine, a simple exchange argument shows that scheduling the jobs in order of non-increasing ratios w_j/p_j gives the optimal schedule [15]. Greedily scheduling the jobs in this order on parallel machines is known as WSPT rule, weighted shortest processing times first, 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 [8]. The computational tractability of the problem was finally settled by showing the existence of a PTAS [14], given that the problem is strongly NP-complete if m is part of the input [3,4].

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 $\mathbb{E}\left[P_j\right]$, 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 the probability that its processing time exceeds t equals

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

We denote this 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 memoryless property, that is,

$$\mathbb{P}\left[P_j > s+t | P_j > s\right] = \mathbb{P}\left[P_j > t\right].$$

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 with the objective to minimize $\mathbb{E}[\sum w_jC_j]$, the analogue of Smith's rule is greedily scheduling the jobs in order of non-increasing ratios $w_j/\mathbb{E}\left[P_j\right]$, also called WSEPT (weighted shortest expected processing time first) [12]. For a single machine, this is again optimal [13]. For parallel machines, it has been shown that the WSEPT rule achieves a performance bound of $(2 - 1/m)$ within the class of all non-anticipatory stochastic scheduling policies [11]. Here, the considered metric is the expected performance of WSEPT relative to that of an (unknown) optimal non-anticipatory scheduling policy. We refer to [10] for the

[*] Corresponding author.
E-mail addresses: c.j.jagtenberg@cwi.nl (C. Jagtenberg), uwe.schwiegelshohn@udo.edu (U. Schwiegelshohn), m.uetz@utwente.nl, marc.uetz@gmail.com (M. Uetz).

0167-6377/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.orl.2013.08.001

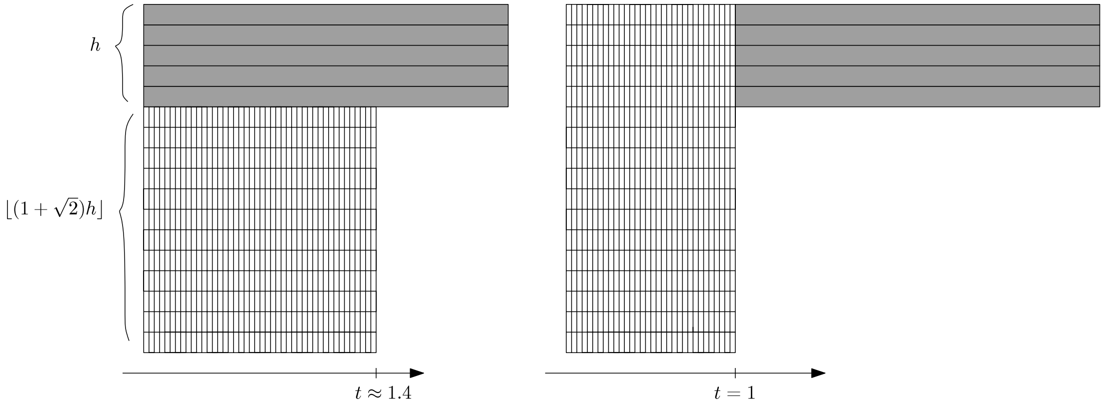

Fig. 1. Two different WSPT schedules, one with optimal objective value v^* on the left, and one with suboptimal value v on the right, respectively.

precise definition on non-anticipatory stochastic scheduling policies. For the purpose of this paper, it suffices to know that non-anticipatory stochastic scheduling policies 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, thus particularly expected processing times $\mathbb{E}\left[P_j\right]$ are even available beforehand.

The major purpose of this paper is to establish the first lower bound for the $(2 - 1/m)$ performance guarantee of [11] 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 [16, Ex. 3.5.12]. Our main result is the following.

Theorem 1. *When scheduling jobs with exponentially distributed processing times on parallel, identical machines in order to minimize* $\mathbb{E}[\sum w_j C_j]$*, the performance guarantee of Smith's rule is no better than* α *with* $\alpha > 1.243$.

To obtain our result, we carefully adapt and analyse the worst-case instance of [8]. Note that the originality of this result lies in the fact that $1.243 > \frac{1}{2}(1+\sqrt{2}) \approx 1.207$. Hence, stochastic scheduling with exponentially distributed processing times has worse worst-case instances than deterministic scheduling. This result may seem counterintuitive, as Pinedo correctly claims the following.

> "It is intuitively acceptable that a deterministic problem may be NP-hard while its counterpart with exponentially distributed processing times allows for a very simple policy to be optimal" [12].

An example for this intuition is given by the problem to minimize the makespan on parallel identical machines: while the problem is NP-hard in deterministic scheduling, the version with exponentially distributed processing times is solved optimally by the LEPT policy (longest expected processing times first) [17]. For the minsum objective considered in this paper, the picture is as follows. For unit weights where $w_j = 1$, the SPT rule is optimal for minimizing $\sum_j C_j$ in the deterministic setting [12], and also SEPT (shortest expected processing time first) is optimal for minimizing $\mathbb{E}[\sum_j C_j]$ 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 $w_1 \geq \cdots \geq w_n$ and $w_1\lambda_1 \geq \cdots \geq w_n\lambda_n$, scheduling the jobs in order $1, 2, \ldots, n$ is optimal [7], while the corresponding deterministic problem is 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, under the realm of minimizing expected performance. Our result shows that with arbitrary weights, the situation is different. Next to this qualitatively new insight, our analysis also sheds light on phenomena in stochastic scheduling which are interesting on their own.

The paper is organized as follows. In Section 2, we briefly review and visualize the worst-case instance presented in [8]. We explain the intuition behind the stochastified instance of [8] in Section 3. Then we derive four technical lemmas about scheduling jobs with exponentially distributed processing times, and finally prove the claimed lower bound for the performance of Smith's rule. Finally, Section 4 contains our conclusions.

2. Recap of the Kawaguchi and Kyan instance

We briefly summarize the instance from [8] that achieves the bound $(1 + \sqrt{2})/2$ for deterministic scheduling, as the instance we propose 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

$$m = h + \lfloor(1+\sqrt{2})h\rfloor$$
$$n = mk + h$$
$$p_j = w_j = 1/k \quad \text{for } 1 \leq j \leq mk$$
$$p_j = w_j = 1 + \sqrt{2} \quad \text{for } mk + 1 \leq j \leq mk + h.$$

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

Let v^* be the total weighted completion time of a schedule where the long jobs are processed first, and v be the total weighted completion time of a schedule in which all short jobs are processed first. Fig. 1 depicts these two schedules. The schedule on the left of Fig. 1 has objective value v^*. Here the last jobs of length $1/k$ finish at time $1 + h/\lfloor(1+\sqrt{2})h\rfloor \approx 1.4$ (for large values of h and k). The schedule on the right of Fig. 1 has value v, and it finishes the last jobs of length $1/k$ exactly at time 1. In Fig. 1 we used $h = 5$ and $k = 32$. It can be verified (see [8]) that $v = (1+\sqrt{2})(2+\sqrt{2})h + (m/2)(1 + 1/k)$ and $v^* = (1+\sqrt{2})^2 h + (m/2)(m/\lfloor(1+\sqrt{2})h\rfloor + 1/k)$. The ratio v/v^* then tends to $(1+\sqrt{2})/2$ as $h \to \infty$ and $k \to \infty$.

3. The stochastic Kawaguchi and Kyan instance

We find it particularly instructive to consider the stochastic analogue of the instance presented by Kawaguchi and Kyan [8],

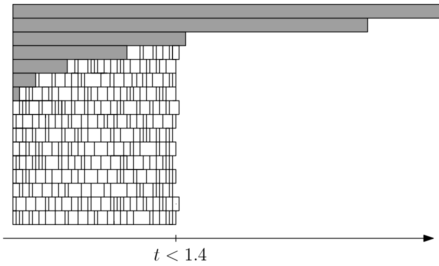

Fig. 2. Schedule with value v^*: all long jobs start at time 0, yet some of these machines are expected to become available for processing short jobs.

even though other instances might lead to comparable results. That said, 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 $P_j \sim \exp(k)$. So the expected processing times of long and short jobs are identical to the deterministic processing times in the worst case example in [8].

The crucial insight when stochastifying the instance by Kawaguchi and Kyan is the following. The non-optimal schedule with value v is essentially identical to the expected situation in stochastic scheduling. However, we will argue that the optimal schedule with value v^* will have a significantly different expected realization with exponentially distributed processing times. We start by sketching the main differences between the deterministic schedules and the expected stochastic schedules in Section 3.1. Then in Section 3.2 we derive some technical lemmas about the behaviour of jobs with exponentially distributed processing times, and finish the analysis in Section 3.3.

3.1. Intuition of the analysis

Suppose we start all long jobs first and greedily fill up the remaining machines with short jobs. As we will formally prove in Lemma 1, we expect the ith long job to finish at time

$$t_i = \sum_{j=1}^{i} \frac{1+\sqrt{2}}{h-j+1}.$$

For a given finite number of machines, the schedule will look like depicted in Fig. 2. The crucial point is that the average expected time that 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 does not matter for the short jobs. Hence, the overall contribution of the short jobs will decrease when compared to the deterministic case, while the contribution of long jobs remains exactly the same.

Suppose on the other hand that we first start all short jobs. The set of short jobs is not likely to produce the ideal rectangle as it did in the deterministic case. However, 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 made arbitrarily small, by letting k, the inverse of the expected processing time of short jobs, be large. In this situation, the expected cost of the schedule is almost the same as the cost in the deterministic case. This is illustrated in Fig. 3.

In other words, in the stochastic setting the performance guarantee of WSEPT deteriorates because the expected value for the optimal policy (long jobs first) decreases in comparison to the deterministic case, while the expected value for the suboptimal policy (short jobs first) remains almost the same.

3.2. Preliminaries for memoryless jobs

In order to formalize the idea from Section 3.1, we first state some technical observations which are needed later in the analysis.

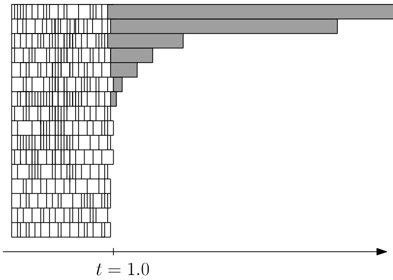

Fig. 3. Schedule with value v: long jobs scheduled only after short jobs, yet expected to start at almost equal times.

Here, λ is an arbitrary positive parameter. We denote by

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

the nth harmonic number, where we define $H_0 := 0$. 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 $h \leq m$ jobs on m machines with i.i.d. exponential processing times $P_j \sim \exp(\lambda)$, the expected number of machines that are idle at a given time t, denoted $m(t)$, is bounded as follows,*

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

Proof. The first completion time is distributed as the minimum of h independent $\exp(\lambda)$ distributions. This is an $\exp(h\lambda)$ distribution, hence it is expected at time $t_1 = \frac{1}{h\lambda}$. After the first job completion, we have $h-1$ jobs remaining. Since the exponential distribution is memoryless, the next completion is expected a time $\frac{1}{(h-1)\lambda}$ later, so $t_2 = \frac{1}{h\lambda} + \frac{1}{(h-1)\lambda}$. By continuing this argument we find that the ith job completion is expected at time

$$t_i = \sum_{j=1}^{i} \frac{1}{(h-j+1)\lambda} = \frac{1}{\lambda} \sum_{j=h-i+1}^{h} \frac{1}{j} = \frac{1}{\lambda}(H_h - H_{h-i}). \tag{1}$$

We now use that $H_i - \ln(i)$ is positive and monotonically decreasing in i [9]. Hence we may conclude that

$$t_i \leq \frac{1}{\lambda}(\ln(h) - \ln(h-i)) = \frac{1}{\lambda} \ln\left(\frac{h}{h-i}\right),$$

which yields

$$i \geq (1-e^{-\lambda t_i})h. \tag{2}$$

Note that $m(t_i) = (m-h)+i$, for $i = 1, \ldots, h$, by definition. Hence, (2) yields

$$m(t_i) \geq (m-h) + (1-e^{-\lambda t_i})h \tag{3}$$

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

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

for all $t \geq 0$. □

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

Lemma 2. *Let $s \leq t$ and consider $k(t-s)$ jobs with i.i.d. processing times $P_j \sim \exp(k)$ and weights $w_j = 1/k$, scheduled on a single machine from time s on. Then for all $\varepsilon > 0$ there exists k large enough* *so that*

$$\mathbb{E}\left[\sum_j w_j C_j\right] \le \int_s^t x\,dx + \varepsilon.$$

Proof. Assuming w.l.o.g. that $\frac{1}{k}|(t-s)$, we have expected job completion times at times $s+1/k, s+2/k, \ldots, s+k(t-s)/k = t$. We therefore calculate rather straightforwardly that $\mathbb{E}\left[\sum_j w_j C_j\right] = \frac{1}{2}(t^2-s^2) + \frac{1}{2k}(t-s)$, so for $k \ge \frac{t-s}{2\varepsilon}$ the claim is true. □

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

Lemma 3. *Suppose we first schedule h i.i.d. long jobs with processing times $P_j \sim \exp(\lambda)$ on m machines, where $h \le m$. We then greedily schedule m k i.i.d. short jobs, with processing times $P_j \sim \exp(k)$ and weights $w_j = 1/k$, where k is large. Let v_{short} be the expected weighted sum of completion times of the short jobs. Then for k large enough,*

$$v_{short} \le \int_0^{T'} f(t)\,t\ dt$$

where $f(t) := (m-h) + (1-e^{-\lambda t})h - 1$ and T' is defined so that $\int_0^{T'} f(t)\,dt = m$.

Proof. First, define T as the average expected machine completion time for machines that process short jobs. We know that when scheduling the short jobs greedily, the schedule is expected to look like illustrated in Fig. 2.

We analyse a scheduling policy π that is inferior to greedy scheduling, that is, it yields an expected value for the total weighted completion times of short jobs $v^{\pi}_{short} \ge v_{short}$. The proof then follows by verifying the claimed upper bound for v^{π}_{short}.

We define π as follows. Let $[i]$ be the ith machine that becomes available to execute short jobs, $t_{[i]}$ be the expected time for that to happen, and for simplicity of notation assume that $i = [i]$. We know that $t_i = 0$ for $i = 1, \ldots, m-h$, and $t_{m-h+i} = \sum_{\ell=0}^{i-1} 1/((h-\ell)\lambda)$ for $i = 1, \ldots, h$. Policy π schedules fixed sets of jobs per machine, in the order in which they become available. More precisely, on machine i, we schedule a fixed set J_i of $k(T-t_i)$ short jobs. By definition of T as the average expected machine completion time for machines that process short jobs, we will have run out of short jobs for all machines i with $t_i > T$. For these machines, we therefore redefine $t_i = T$. Policy π is indeed inferior in contrast to greedy scheduling, as it lacks the load balancing towards the end of the schedule. That is, there is positive probability that a machine is left idle although other machines have yet unscheduled jobs, which cannot happen when scheduling the short jobs greedily. Yet note that, by definition, the expected machine completion times equal T for all machines that process short jobs.

By Lemma 2, we know that under π it holds for the short jobs on machine i that

$$\sum_{j \in J_i} w_j C_j \le \int_{t_i}^{T} t\,dt + \varepsilon_i,$$

for any $\varepsilon_i > 0$. Now we sum over all machines, where we let $\varepsilon_i = 0$ for all machines i that become available while there are no more short jobs. We get

$$v^{\pi}_{short} \le \sum_{i=1}^{m} \int_{t_i}^{T} t\,dt + \varepsilon_i = \int_0^T m(t) t\,dt + \varepsilon, \tag{4}$$

where $m(t)$ is defined as the expected number of machines at time t that are available for processing short jobs, and $\varepsilon := \sum_i \varepsilon_i$.

Now $f(t) = (m-h) + (1-e^{-\lambda t})h - 1$, and Lemma 1 yields $m(t) > f(t)$ for all $t \ge 0$. The functions $f(t)$ and $m(t)$ are illustrated in Fig. 4. By definition of T' we have $m = \int_0^T m(t)\,dt = \int_0^{T'} f(t)\,dt$, which implies that the two grey areas in Fig. 4 are equal in size. Also note that $m(t) - f(t)$ is nonnegative for all $t \ge 0$. Therefore,

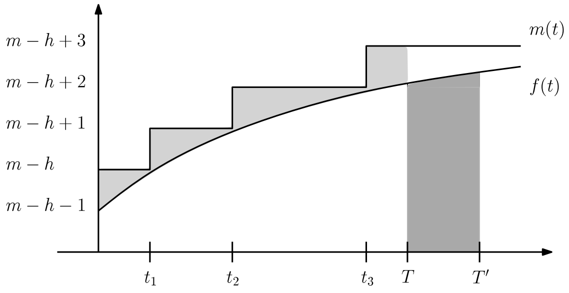

Fig. 4. Illustration of functions $m(t)$, $f(t)$, and values T and T'.

$$\begin{aligned}\int_0^T (m(t)-f(t))t\,dt &< T\int_0^T (m(t)-f(t))\,dt \\ &= T\int_T^{T'} f(t)\,dt \\ &< \int_T^{T'} f(t)t\,dt.\end{aligned}$$

Here, the first inequality follows from $m(t)-f(t) \ge 0$, the equality from $\int_0^T m(t)\,dt = \int_0^{T'} f(t)\,dt$, and the last inequality from $f(T) \ge 0$ and f being monotone non-decreasing. We conclude from the previous inequalities that there exists some constant $\eta > 0$ so that

$$\int_0^T m(t)t\,dt + \eta \le \int_0^{T'} f(t)t\,dt. \tag{5}$$

Therefore, by choosing $\varepsilon \le \eta$, we may conclude from (4) and (5), that

$$v^{\pi}_{short} \le \int_0^T m(t)\,t\,dt + \varepsilon \le \int_0^{T'} f(t)\,t\,dt. \quad \square$$

Intuitively, the expression $\int_0^{T'} f(t)t\,dt$ equals the total weighted completion time for infinitesimally short jobs with total expected processing m, scheduled on "machines" with availability $f(t)$. As $m(t) \ge f(t)$, the actual availability of machines for short jobs is higher. We bound the contribution of the jobs that are processed in the light grey area of Fig. 4 by the contribution they would have if they were processed in the dark grey area.

Finally, the next lemma makes a statement about the machine completion times when scheduling a block of (short) jobs, as illustrated in Fig. 3.

Lemma 4. *Suppose we schedule mk i.i.d. short jobs with processing times $P_j \sim \exp(k)$ greedily on m machines. Then the average expected machine completion time equals 1, and for any $\delta > 0$ there exists k large enough such that the earliest expected machine completion time is at time $t \ge 1-\delta$.*

Proof. The claim about the average expected machine completion time is clear, because the total expected processing is m. For the second claim, consider the first time, say t, that a machine runs out of jobs. We know from Lemma 1 that the last machine that runs out of jobs is expected to be at time $t + \sum_{i=1}^{m-1} \frac{1}{ik}$. For m large enough, we have $\sum_{i=1}^{m-1} \frac{1}{ik} \le \frac{1}{k}[\ln(m) + \gamma]$. Here

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

denotes the Euler–Mascheroni constant [2]. Of course, the average expected machine completion time must be less than the last expected machine completion time. Therefore, we have $1 \leq t + \sum_{i=1}^{m-1} \frac{1}{ik} \leq t + \frac{1}{k}[\ln(m) + \gamma]$. If we now let $k \geq (\ln(m) + \gamma)/\delta$, we get $1 \leq t + \delta$. □

3.3. Lower bound on performance of Smith's rule

Let v^* denote the expected objective value $\mathbb{E}\left[\sum_j w_j C_j\right]$ for the policy that first schedules all long jobs. Similarly, let v denote the expected objective value for the policy that starts long jobs only when there is no short job left to be scheduled. Both policies are WSEPT, hence the ratio v/v^* is a lower bound for the approximation ratio of Smith's rule in stochastic machine scheduling with exponentially distributed processing times. We choose h sufficiently large, and k, a multiple of $\lfloor(1+\sqrt{2})h\rfloor$, we may choose arbitrarily large in comparison to h (i.e., $k \gg h$). In fact, we can choose these two parameters in such a way that all our technical lemmas from Section 3.2 do apply.

The optimal policy, v^*. We split v^* up into the contribution of long jobs v^*_{long} and the contribution of short jobs v^*_{short}. So

$$v^* = v^*_{long} + v^*_{short}.$$

The value v^*_{long}: We start all h 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

$$v^*_{long} = h(1+\sqrt{2})^2, \tag{6}$$

which is the same as in the deterministic case.

The value v^*_{short}: Just like in the proof of Lemma 3 denote by $m(t)$ the expected number of machines at time t that is available for processing short jobs, and T be the average expected machine completion time for machines that process short jobs. We now use Lemma 3 where

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

Following the proof of Lemma 3, we need to compute a value $T' \geq T$ large enough so that $\int_0^{T'} f(t)\,dt \geq m$. We have not attempted to solve this analytically, but one can check numerically that for $m = h + \lfloor(1+\sqrt{2})h\rfloor$ and $h \to \infty$,

$$T' = 1.2933 \tag{7}$$

suffices to process the short jobs when machine availabilities are governed by function $f(t)$ rather than the true value $m(t)$. Then v^*_{short}, the expected weighted sum of completion times for all mk short jobs, can be bounded using Lemma 3. We thus find, for h and k sufficiently large,

$$v^*_{short} \leq \int_0^{T'} f(t)t\,dt. \tag{8}$$

With (7) and (8) we can calculate

$$v^*_{short} \leq 2.266h - 0.836. \tag{9}$$

Combining (6) and (9) gives

$$v^* = v^*_{long} + v^*_{short} \leq (1+\sqrt{2})^2 h + 2.266h - 0.836. \tag{10}$$

The worst case policy, v. 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,

$$v = v_{short} + v_{long}.$$

The value v_{short}: We have m machines working on mk jobs with processing times $P_j \sim \exp(k)$. According to Lemma 4, on average a machine is expected to finish with these jobs at time 1, and for any $\delta > 0$, we can find k large enough so that no machine is expected to finish before time $1-\delta$. Hence, the average expected completion time of the set of short jobs on each machine is at least $(1-\delta)/2$. Therefore, for any $\varepsilon > 0$, there is k large enough so that, by choosing $\varepsilon = m\delta$,

$$v_{short} \geq m/2 - \varepsilon/2. \tag{11}$$

The value v_{long}: Remember that the schedule is expected to look like depicted in Fig. 3. Using Lemma 4 again, we know that long jobs are expected to start no earlier than $1-\delta$, for any $\delta > 0$. So by assuming they all start at this time, we get a lower bound for their completion times. 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 h long jobs, for any $\varepsilon > 0$ there is k large enough so that

$$v_{long} \geq (2+\sqrt{2})\,(1+\sqrt{2})h - \varepsilon/2, \tag{12}$$

by choosing $\delta = \varepsilon/(2h(1+\sqrt{2}))$. With (11) and (12) we now have

$$v = v_{short} + v_{long} \geq m/2 + (2+\sqrt{2})\,(1+\sqrt{2})h - \varepsilon. \tag{13}$$

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

$$\alpha \geq \frac{v}{v^*}.$$

Remember that $m = h + \lfloor(1+\sqrt{2})h\rfloor$. Now for carefully chosen $k \gg h$, and taking $h \to \infty$, Eqs. (10) and (13) give

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

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.

Optimizing the parameters. What remains to be done is to optimize over the parameters of the instance to improve the obtained lower bound. To that end, recall that the considered instance has h long jobs and $m = h + \lfloor(1+\sqrt{2})h\rfloor \approx 3.4h$ machines, and long jobs have processing times $P_j \sim \exp(\frac{1}{1+\sqrt{2}}) \approx \exp(0.41)$. However, these parameters are optimized for the deterministic instance. Taking slightly more long jobs, namely by letting $m = 2.3h$, with somewhat shorter processing times, namely $P_j \sim \exp(0.56)$, we obtain a ratio of at least 1.2436, which finally proves Theorem 1.

4. Conclusion

The numerical calculations have been performed using Mathematica. 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 necessarily optimal for the stochastic counterpart with exponentially distributed processing times. In conclusion, improvements in the ratio 1.243 might be possible. Yet, the upper bound $(2 - 1/m)$ seems out of reach. This leaves the question to improve the upper bound on the performance guarantee for WSEPT; in that respect, it is interesting to note that the analysis of [11] does not explicitly exploit the exponential distribution; it is valid in more generality.

Acknowledgements

A preliminary version of this paper appeared in the proceedings of the 8th Workshop on Approximation and Online Algorithms [6]. The second and the third author thank the organizers of the 2010 Dagstuhl Seminar on Scheduling, which was a starting point for this work. We also thank an anonymous referee for helpful comments.

References

[1] J.L. Bruno, P.J. Downey, G.N. Frederickson, Sequencing tasks with exponential service times to minimize the expected flowtime or makespan, Journal of the Association for Computing Machinery 28 (1981) 100–113.

[2] L. Euler, De progressionibus harmonicis observationes, Originally published in: Commentarii Academiae Scientiarum Petropolitanae, Vol. 7, 1740, pp. 150–161.

[3] M.R. Garey, D.S. Johnson, Strong NP-completeness results: motivation, examples, and implications, Journal of the ACM 25 (1978) 499–508.

[4] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, New York, 1979.

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

[6] C. Jagtenberg, U. Schwiegelshohn, M. Uetz, Lower bounds for Smith's rule in stochastic machine scheduling, in: K. Jansen, R. Solis-Oba (Eds.), Approximation and Online Algorithms, in: Lecture Notes in Computer Science, vol. 6534, 2011, pp. 142–153.

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

[8] T. Kawaguchi, S. Kyan, Worst case bound on an LRF schedule for the mean weighted flow-time problem, SIAM Journal on Computing 15 (1986) 1119–1129.

[9] D. Knuth, The Art of Computer Programming. Volume 1: Fundamental Algorithms, third ed., Addison-Wesley, 1997, pp. 75–79. Section 1.2.7: Harmonic Numbers.

[10] R.H. Möhring, F.J. Radermacher, G. Weiss, Stochastic scheduling problems I: general strategies, ZOR—Zeitschrift für Operations Research 28 (1984) 193–260.

[11] R.H. Möhring, A.S. Schulz, M. Uetz, Approximation in stochastic scheduling: the power of LP-based priority policies, Journal of the Association for Computing Machinery 46 (1999) 924–942.

[12] M. Pinedo, Scheduling: Theory, Algorithms, and Systems, second ed., Prentice-Hall, Upper Saddle River, NJ, 2002.

[13] M.H. Rothkopf, Scheduling with random service times, Management Science 12 (1966) 703–713.

[14] M. Skutella, G.J. Woeginger, A PTAS for minimizing the total weighted completion time on identical parallel machines, Mathematics of Operations Research 25 (2000) 63–75.

[15] W.E. Smith, Various optimizers for single-stage production, Naval Research Logistics Quarterly 3 (1956) 59–66.

[16] M. Uetz, Algorithms for deterministic and stochastic scheduling, Ph.D. Thesis, Institut für Mathematik, Technische Universität Berlin, Germany, 2001. Published by Cuvillier Verlag, Göttingen, Germany, 2002.

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