On Index Policies for Stochastic Minsum Scheduling

Franziska Eberle[a,*], Felix Fischer[b], Jannik Matuschke[c], Nicole Megow[a]

[a] *Department of Mathematics and Computer Science, University of Bremen, Germany*
[b] *School of Mathematical Sciences, Queen Mary University of London, UK*
[c] *Faculty of Economics and Business, KU Leuven, Belgium*

Abstract

Minimizing the sum of completion times when scheduling jobs on m identical parallel machines is a fundamental scheduling problem. Unlike the well-understood deterministic variant, it is a major open problem how to handle stochastic processing times. We show for the prominent class of index policies that no such policy can achieve a distribution-independent approximation factor. This strong lower bound holds even for simple instances with deterministic and two-point distributed jobs. For such instances, we give an $O(m)$-approximative list scheduling policy.

Keywords: stochastic scheduling, parallel machines, total completion time, approximation algorithm

1. Introduction

It is a classical and well-understood problem to schedule jobs on identical parallel machines with the objective of minimizing the sum of completion times. In this problem, we are given a set of jobs $J = \{1, \dots, n\}$, where each job $j \in J$ has a processing time p_j that indicates for how many time units it has to be processed non-preemptively on one of the m given machines. At any point in time, a machine can process at most one job. The objective is to find a schedule that minimizes the total completion time, $\sum_{j \in J} C_j$, where C_j denotes the completion time of job j. This problem is denoted as $P||\sum C_j$ in the standard three-field notation [11]. It is well-known that scheduling the jobs as early as possible in *Shortest Processing Time* (SPT) order solves the problem optimally on a single [27] as well as on multiple machines [3].

Stochastic scheduling. Uncertainty in the processing times is ubiquitous in many applications. Although the first results on scheduling with probabilistic information date back to the 1960s, the question how to schedule jobs with stochastic processing times is hardly understood.

We investigate a stochastic variant of the minsum scheduling problem. The processing time of a job j is modeled by a random variable P_j with known probability distribution. We assume that the processing time distributions for individual jobs are independent. The objective is to find a non-anticipatory *scheduling policy* Π that decides for any time $t \geq 0$ which jobs to schedule. A non-anticipatory policy has to base these scheduling decisions only on observed information up to time t and/or on a priori knowledge about the distributions. In particular, the policy is not allowed to use information about the actual realizations of processing times of jobs that have not yet started by time t. For a more thorough introduction to non-anticipatory scheduling policies see [17, 20, 25].

For a non-anticipatory policy Π, the value of the objective function $\sum_j C_j^\Pi$ is a random variable. A natural generalization of $P||\sum C_j$ is to ask for minimizing the *expected value* of this random variable, i.e., to minimize $\sum_j \mathbb{E}[C_j^\Pi]$ by linearity of expectation. We drop the superscript whenever it is clear from the context. This stochastic scheduling problem is commonly denoted by $P||\mathbb{E}[\sum C_j]$.

List scheduling and index policies. An important class of policies in (stochastic) scheduling is *list scheduling* [10]. A list scheduling policy maintains a (static or dynamic) priority list of jobs and schedules at any time as many available jobs as possible in the order given by the list. The aforementioned SPT rule falls into this class. List scheduling policies are the simplest type of *elementary* policies, that is, policies that start jobs only at the completion times of other jobs (or at time 0). For further details on the classification of (non-preemptive) stochastic scheduling policies we refer to [20, 21].

A prominent subclass of list scheduling policies is called *index policies* [6, 28]. An index policy assigns a priority index to each unfinished job, where the index for a job is determined by the (distributional) parameters and state of the job itself but independent of other jobs. If job preemption is not allowed, then these priority indices are static, that is, they do not change throughout the execution of the scheduling policy. Moreover, index policies assign jobs with the same probability distribution the same priority index and do not take the number of jobs or the number of machines into account.

*Corresponding author
Email addresses: feberle@uni-bremen.de (Franziska Eberle), felix.fischer@qmul.ac.uk (Felix Fischer), jannik.matuschke@kuleuven.be (Jannik Matuschke), nicole.megow@uni-bremen.de (Nicole Megow)

Concerning stochastic minsum scheduling, a natural generalization of the SPT rule, the *Shortest Expected Processing Time* (SEPT) rule, plays an important role. This index policy is optimal for minsum scheduling on a single machine [23]. This is true also for the weighted setting and the weighted SEPT rule, WSEPT. Moreover, SEPT is known to be optimal for $P||\mathbb{E}[\sum C_j]$ when processing times follow exponential distributions [1], geometric distributions [8], or if the processing time distributions are pairwise stochastically comparable [29].

Other index policies that perform provably well for certain stochastic scheduling settings, are, e.g., the Longest Expected Processing Time (LEPT) [30] and Largest Variance First (LVF) [22] rules, and the Gittins Index [5]. For an overview on theory and applications of index policies (with a focus on interruptible jobs) we refer to [7, 9].

Further related results. For arbitrary instances of $P||\mathbb{E}[\sum C_j]$, there is no optimal policy known. In the past decade, research has focused on approximative policies. A stochastic scheduling policy Π is an α*–approximation*, for $\alpha \geq 1$, if for all problem instances I it holds that $\sum_{j\in J_I} \mathbb{E}[C_j^{\Pi}] \leq \alpha \sum_{j\in J_I} \mathbb{E}[C_j^*]$. Here, C_j^* denotes the completion time under an optimal non-anticipatory scheduling policy on the given instance I.

Starting with the seminal paper [19], several scheduling policies have been developed for this problem (with arbitrary job weights) and generalizations such as precedence constraints [26], heterogeneous machines [12, 25] and online models [12, 17, 24]. In all cases, the approximation guarantee depends on the probability distributions of the processing times. More precisely, the guarantee is in the order $\mathcal{O}(\Delta)$ where Δ is an upper bound on the squared coefficients of variation of the processing time distributions P_j, that is, $\mathrm{Var}[P_j]/\mathbb{E}[P_j]^2 \leq \Delta$ for all jobs j.

Besides linear programming relaxations, the (W)SEPT policy plays a key role in the aforementioned results. This index policy, being optimal on a single machine, has been studied extensively as a promising candidate for approximating $P||\mathbb{E}[\sum C_j]$ well. Recently, the upper bound for WSEPT has been decreased to $1+(\sqrt{2}-1)/2\cdot(1+\Delta)$ [14]. On the negative side, it has been shown independently that neither WSEPT [15] nor SEPT [2, 13] can achieve approximation factors independent of Δ, when there are many machines.

A remarkable recent result is a list scheduling policy for $P||\mathbb{E}[\sum C_j]$ with the first distribution-independent approximation factor of $\mathcal{O}(m \log n + \log^2 n)$ [13]. This policy is based on SEPT but in addition it takes carefully into account the probability that a job turns out to be long.

Nevertheless, it remains a major open question whether there is a constant factor approximation for this problem even if all weights are equal. Interestingly, there is an index policy with an approximation factor 2 for the *preemptive* (weighted) variant of our stochastic scheduling problem [18]. It is natural to ask whether index policies can achieve a constant approximation factor also in the non-preemptive setting.

Our contribution. As our main result, we rule out any constant or even distribution-independent approximation factor for index policies. More precisely, we give a lower bound of $\Omega(\Delta^{1/4})$ for any index policy for $P||\mathbb{E}[\sum C_j]$. This strong lower bound implies that prioritizing jobs only according to their individual processing time distribution cannot lead to better approximation factors. More sophisticated policies are needed that take the entire job set and the machine setting into account. Somewhat surprisingly, our lower bound holds even for very simple instances with only two types of jobs, identical deterministic jobs and a set of stochastic jobs that all follow the same two-point distribution. For this class of instances we provide an alternative list scheduling policy—carefully taking the number of jobs and machines into account—that is an $\mathcal{O}(m)$-approximation.

2. Lower bound for index policies

In this section we prove our main result, a distribution-dependent lower bound on the approximation factor that any index policy can achieve.

Theorem 1. *Any index policy has an approximation factor* $\Omega(\Delta^{1/4})$ *for* $P||\mathbb{E}[\sum_j C_j]$.

To prove this lower bound we consider a simple class of instances that we call *Bernoulli-type instances.* This class consists of two types of jobs, deterministic jobs J_d and stochastic jobs J_s, with jobs of each type following the same distribution. A deterministic job $j \in J_d$ has processing time $P_j = p$, and a stochastic job $j \in J_s$ has processing time $P_j = 0$ with probability $q \in (0,1)$ and $P_j = l > 0$ with probability $1-q$.

For the stochastic jobs, i.e., $j \in J_s$, let $X_j = \mathbf{1}_{\{P_j = l\}}$. Then, X_j is a Bernoulli-distributed random variable that indicates if $j \in J_s$ is long. As the processing time variables P_j are independent, the same holds for $X_j, j \in J_s$. Hence, $X := \sum_{j\in J_s} X_j$ follows a Binomial distribution with success probability q and size parameter $n_s := |J_s|$, denoted by $X \sim \mathrm{Bin}(n_s, q)$, with expected value $\mathbb{E}[X] = n_s \cdot q$. Intuitively, X counts the number of jobs that turn out to be long.

In the proof, we rely on the following concentration result for Bernoulli variables, which is a variant of the Chernoff-Hoeffding bound [16].

Lemma 2. *For* $1 \leq i \leq n$ *let* X_i *be independent, identically distributed Bernoulli variables and let* $X := \sum_{i=1}^n X_i$. *For* $0 < \varepsilon < 1$*, it holds*

1. $\mathbb{P}[X \geq (1+\varepsilon)\mathbb{E}[X]] \leq \exp(-\varepsilon^2\,\mathbb{E}[X]/3)$ *and*
2. $\mathbb{P}[X \leq (1-\varepsilon)\mathbb{E}[X]] \leq \exp(-\varepsilon^2\,\mathbb{E}[X]/2)$.

Proof of Theorem 1. We define two families of Bernoulli-type instances, $I_1(\Delta, m)$ and $I_2(\Delta, m)$, for the problem $\mathrm{P}||\mathbb{E}[\sum C_j]$ where Δ is the upper bound on $\mathrm{Var}[P_j]/\mathbb{E}[P_j]^2$. The instances differ only in the number of deterministic and stochastic jobs, n_d and n_s, but not in the processing time distributions. We define the processing time for deterministic jobs in J_d to be $p = 1$, and for stochastic jobs $j \in J_s$ we define

$$P_j = \begin{cases} 0 & \text{with probability } 1 - 1/\Delta \\ \Delta^{3/2} & \text{with probability } 1/\Delta. \end{cases}$$

Note that $\mathbb{E}[P_j] = \Delta^{1/2}$ and $\mathbb{V}ar[P_j] = \Delta^2 - 1$ for $j \in J_s$. Hence, the squared coefficients of variation, $\mathrm{Var}[P_j]/\mathbb{E}[P_j]^2$, are indeed bounded from above by Δ.

For such Bernoulli-type instances there are only two index policies, one where the deterministic jobs have higher priority, denoted by $J_d \prec J_s$, and one where the stochastic jobs have higher priority, denoted by $J_s \prec J_d$. We show that for any fixed $\Delta > 1$, there exists a value of m such that the cost of the schedule produced by $J_d \prec J_s$ on instance $I_1(\Delta, m)$ is greater by a factor of $\Omega(\Delta^{1/4})$ than the cost of the schedule produced by $J_s \prec J_d$, and vice versa for instance $I_2(\Delta, m)$. As the instances $I_1(\Delta, m)$ and $I_2(\Delta, m)$ are indistinguishable by a index policy, this result implies the lower bound.

The first instance. Instance $I_1(\Delta, m)$ is defined by $n_d = \Delta^{3/4} m$ and $n_s = \frac{1}{2}\Delta m$; w.l.o.g. we assume that $n_d/m \in \mathbb{Z}$. We distinguish both priority orders.

Deterministic jobs before stochastic jobs. When the deterministic jobs in J_d are scheduled first, then no job in J_s starts before n_d/m. Thus,

$$\mathbb{E}\Big[\sum_{j\in J} C_j\Big] \geq \frac{n_d}{m} n_s = \frac{1}{2}\Delta^{7/4} m.$$

Stochastic jobs before deterministic jobs. Let X be the random variable counting the number of jobs in J_s that turn out to be long. Then, $X \sim \mathrm{Bin}(n_s, 1/\Delta)$ and $\mathbb{E}[X] = m/2$. We distinguish two cases based on the value of X.

$X \leq 3/4m$. Every stochastic job starts at time 0. Thus,

$$\mathbb{E}\left[\sum_{j\in J_s} C_j \,|\, X \leq \frac{3}{4}m\right] \leq \frac{3}{4}\Delta^{3/2} m.$$

Furthermore, at least $m/4$ machines are free for scheduling deterministic jobs, J_d, at total cost bounded by

$$\mathbb{E}\left[\sum_{j\in J_d} C_j \,\middle|\, X \leq \frac{3}{4}m\right] \leq \frac{n_d(n_d+1)}{\frac{1}{4}m} \leq 8\Delta^{3/2} m.$$

$X > 3/4m$. We get a (very crude) upper bound on the expected cost by assuming all jobs have processing time $\Delta^{3/2}$ and then scheduling them on a single machine:

$$\begin{aligned}\mathbb{E}\left[\sum_{j\in J} C_j \,\middle|\, X > \frac{3}{4}m\right] &< \frac{1}{2}(n_d+n_s)(n_d+n_s+1)\Delta^{3/2} \\ &\leq 3\Delta^{7/2} m^2.\end{aligned}$$

We use Lemma 2 to bound the probability of the second case, that is, $\mathbb{P}[X > 3/4m] < \exp(-m/24)$. Using the law of total expectation, we get

$$\begin{aligned}\mathbb{E}\Big[\sum_{j\in J} C_j\Big] &\leq \mathbb{P}\Big[X \leq \frac{3}{4}m\Big]\mathbb{E}\Big[\sum_{j\in J} C_j \,\Big|\, X \leq \frac{3}{4}m\Big] \\ &\quad + \mathbb{P}\Big[X > \frac{3}{4}m\Big]\mathbb{E}\Big[\sum_{j\in J} C_j \,\Big|\, X > \frac{3}{4}m\Big] \\ &\leq \frac{3}{4}\Delta^{3/2} m + 8\Delta^{3/2} m + \exp\left(-\frac{m}{24}\right)\cdot 3\Delta^{7/2} m^2 \\ &= \mathcal{O}(\Delta^{3/2} m),\end{aligned}$$

for sufficiently large m. Thus, on sufficiently many machines, the index policy $J_d \prec J_s$ has total cost greater by a factor of $\Omega(\Delta^{1/4})$ than the cost of policy $J_s \prec J_d$.

The second instance. Instance $I_2(\Delta, m)$ is defined by $n_d = \Delta^{5/4} m$ and $n_s = 2\Delta m$. Let X again denote the number of jobs in J_s that turn out to be long. Then, $X \sim \mathrm{Bin}(2\Delta m, 1/\Delta)$ and hence, $\mathbb{E}[X] = 2m$. We analyze both index policies.

Deterministic jobs before stochastic jobs. We condition on two events regarding the realized value of X.

$X \leq 3m$: Every machine is assigned at most $n_d/m = \Delta^{5/4}$ deterministic jobs and at most three long stochastic jobs. Hence, every (stochastic) job has completed by time $\Delta^{5/4} + 3\Delta^{3/2}$. Thus,

$$\begin{aligned}\mathbb{E}\left[\sum_{j\in J} C_j \,\middle|\, X \leq 3m\right] &\leq \frac{n_d^2}{m} + \left(\Delta^{5/4} + 3\Delta^{3/2}\right) n_s \\ &= O(\Delta^{5/2} m)\end{aligned}$$

$X > 3m$: Lemma 2 implies that $\mathbb{P}[X > 3m] \leq \exp(-m/6)$. Using again the fact that scheduling all jobs on one machine and assuming $P_j = \Delta^{3/2}$ for $j \in J$ is an upper bound, we have

$$\mathbb{E}\left[\sum_{j\in J} C_j \,\middle|\, X > 3m\right] \leq 3\Delta^{7/2} m^2.$$

With the law of total expectation, we get

$$\mathbb{E}\left[\sum_{j\in J} C_j\right] = O(\Delta^{5/2} m).$$

Stochastic jobs before deterministic jobs. Here, we condition on the event that $X > m$, where X is the random variable counting the number of long stochastic jobs.

***X* > *m*:** Lemma 2 implies that $\mathbb{P}[X \leq m] \leq \exp(-m/4)$. Hence, $\mathbb{P}[X > m] \geq 1/2$ for $m \geq 4$. If $X > m$, then every machine receives at least one stochastic job before it starts processing the first deterministic job. Thus,

$$\mathbb{E}\left[\sum_{j\in J_d} C_j \,\middle|\, X > m\right] \geq \Delta^{3/2} n_d = \Delta^{11/4}.$$

With the law of total expectation we conclude that

$$\mathbb{E}\left[\sum_{j\in J} C_j\right] \geq \frac{1}{2}\mathbb{E}\left[\sum_{j\in J} C_j \,\middle|\, X > m\right] = \Omega(\Delta^{11/4}).$$

Thus, on sufficiently many machines, the index policy $J_s \prec J_d$ has total cost greater by a factor of $\Omega(\Delta^{1/4})$ than the cost of policy $J_d \prec J_s$.

In summary, we have provided two instances $I_1(\Delta, m)$ and $I_2(\Delta, m)$ which are indistinguishable by any index policy. We have shown that, on the one hand, the policy $J_d \prec J_s$ has total expected cost greater by a factor of $\mathcal{O}(\Delta^{1/4})$ than the policy $J_s \prec J_d$ for the first instance $I_1(\Delta, m)$. On the other hand, the total expected cost of the policy $J_s \prec J_d$ is greater by a factor of $\Omega(\Delta^{1/4})$ than $J_d \prec J_s$ on the second instance $I_2(\Delta, m)$. Thus, the approximation ratio of any index policy is lower bounded by $\Omega(\Delta^{1/4})$. □

3. Upper bound for Bernoulli-type instances

We show that taking the number of machines and jobs into account allows for a list scheduling policy that is $\mathcal{O}(m)$-approximative for the class of Bernoulli-type instances considered in the previous section.

Theorem 3. *There exists an $\mathcal{O}(m)$-approximative list scheduling policy for Bernoulli-type instances of* $\mathrm{P}||\mathbb{E}[\sum C_j]$.

By rescaling, we can assume w.l.o.g. that deterministic jobs $j \in J_d$ have processing time $P_j = p$ and stochastic jobs $j \in J_s$ have processing time

$$P_j = \begin{cases} 0 & \text{with probability } 1 - 1/l \\ l & \text{with probability } 1/l, \end{cases}$$

where $l > 0$.

Regarding the total scheduling cost of any policy, we observe the following.

Observation 4. *Individually scheduling J_d or J_s on m machines starting at time 0 gives a lower bound on the cost of an optimal policy. We denote these job-set individual scheduling cost by $\sum_{j\in J_t} \mathbb{E}[C_j^0]$ where $t \in \{s, d\}$. Obviously, the sum of both also is a lower bound on the optimum cost,*

$$\sum_{j\in J} C_j^* \geq \sum_{j\in J_d} \mathbb{E}[C_j^0] + \sum_{j\in J_s} \mathbb{E}[C_j^0].$$

For deterministic jobs, the job-set individual scheduling cost can easily be bounded by an averaging argument:

$$\sum_{j\in J_d} \mathbb{E}[C_j^0] \geq \frac{n_d(n_d+1)}{2m} \cdot p. \tag{1}$$

We prove the main result of this section, the existence of an $\mathcal{O}(m)$-approximation, through a careful analysis of the relation between the parameters of a Bernoulli-type instance. In Lemma 5 we consider the case of few deterministic jobs before analyzing the case that there are less stochastic jobs than deterministic ones in Lemma 6. In Lemma 7, we make a useful observation on X, the random variable counting the stochastic jobs with long processing time. This observation is the basis for Lemma 8 which handles the remaining cases.

Firstly, note that in case of few deterministic jobs, $J_s \prec J_d$ is an $\mathcal{O}(1)$-approximation.

Lemma 5. *$J_s \prec J_d$ is a 2-approximation for Bernoulli-type instances satisfying $n_d \leq m$.*

Proof. The cost of scheduling $J_s \prec J_d$ is at most the cost of J_s and the cost of one deterministic job per machine starting at the completion of the last stochastic job on that machine. Then, by linearity of expectation,

$$\begin{aligned}\sum_{j\in J} \mathbb{E}[C_j] &= \sum_{j\in J_s} \mathbb{E}[C_j^0] + \sum_{j\in J_d} \mathbb{E}[S_j + p] \\ &\leq 2\sum_{j\in J_s} \mathbb{E}[C_j^0] + n_d p \\ &\leq 2\sum_{j\in J} \mathbb{E}[C_j^*].\end{aligned}$$ □

Moreover, if there are less stochastic jobs than deterministic ones, $J_d \prec J_s$ is $\mathcal{O}(1)$-approximative.

Lemma 6. *$J_d \prec J_s$ is a 5-approximation for Bernoulli-type instances with $n_d > m$ and $n_s \leq 2n_d$.*

Proof. When scheduling in order $J_d \prec J_s$, machines start processing jobs in J_s no later than $\lceil \frac{n_d}{m} \rceil p \leq 2\frac{n_d}{m} p$ when all jobs in J_d have completed. Thus, the total cost of scheduling J_s after J_d is

$$\sum_{j\in J_s} \mathbb{E}[C_j^0] + n_s \cdot 2\frac{n_d}{m} p \leq \sum_{j\in J_s} \mathbb{E}[C_j^0] + 4\sum_{j\in J_d} \mathbb{E}[C_j^0],$$

which follows from (1). Adding the total cost of the deterministic jobs J_d implies the 5-approximation. □

To handle the remaining instances, recall X, the random variable counting the number of stochastic jobs that turn out to be long. Furthermore, fix a sequence of the stochastic jobs J_s and let Z_i denote the position of the ith long job in that sequence. The following lemma states some elementary properties of Z_i. For the sake of completeness, we give a proof in the appendix.

Lemma 7. *For X and Z_i defined as before and $1 \le i \le \lambda m, k \le n_s$ for $\lambda \in \{1, \dots, \lfloor \frac{n_s}{m} \rfloor\}$, the following holds:*

(i) $\mathbb{E}[Z_i \mid X = k] = \frac{i}{k+1}(n_s + 1)$.
(ii) $\mathbb{E}[Z_i \mid \lambda m \le X < (\lambda+1)m] \le \frac{i}{\lambda m+1}(n_s + 1)$.
(iii) $\mathbb{E}[n_s - \Pi_m \mid m \le X < 2m] \ge \frac{n_s}{4m}$.

With the previous lemma, we are now ready to analyze the remaining cases, i.e., instances with twice as many stochastic jobs as deterministic ones and more deterministic jobs than machines.

Lemma 8. *$J_s \prec J_d$ is an $\mathcal{O}(m)$-approximation for Bernoulli-type instances with $n_s > 2n_d > 2m$.*

Proof. We analyze the performance of $J_s \prec J_d$ by conditioning on the number X of long jobs.

$\boldsymbol{0 \le X < m}$: Let $0 \le k < m$ and consider all realizations such that $X = k$. Then, there exist at least $m - k$ machines that do not schedule stochastic jobs. Hence,

$$\mathbb{E}\left[\sum_{j \in J} C_j \,\middle|\, X = k\right] \le k \cdot l + \frac{n_d(n_d+1)}{2(m-k)} p$$
$$\le k \cdot l + n_d^2 p.$$

The optimal policy also has to process the k long stochastic jobs and due to (1) applied on $m - k$ machines it follows

$$\mathbb{E}\left[\sum_{j \in J} C_j^* \,\middle|\, X = k\right] \ge k \cdot l + \frac{n_d^2 p}{2m},$$

where C_j^* again denotes the completion time of j in an optimal schedule. Thus,

$$\mathbb{E}\left[\sum_{j \in J} C_j \,\middle|\, X = k\right] \le 2m \mathbb{E}\left[\sum_{j \in J} C_j^* \mid X = k\right].$$

$\boldsymbol{\lambda m \le X < (\lambda+1)m}$ **for** $\boldsymbol{\lambda \in \{1, \dots, \lfloor \frac{n_s}{m} \rfloor\}}$: All stochastic jobs are finished at the latest by $(\lambda+1)\,l$. Hence, from time $(\lambda+1)\,l$ on, all machines process deterministic jobs only. Thus,

$$\sum_{j \in J} \mathbb{E}[C_j \mid \lambda m \le X < (\lambda+1)m]$$
$$\le \sum_{j \in J} \mathbb{E}[C_j^0 \mid \lambda m \le X < (\lambda+1)m] + (\lambda+1) l n_d. \quad (2)$$

As noted in Observation 4, the first term is a lower bound on the optimum cost and it remains to bound the second term, i.e., $(\lambda+1) l n_d$.

Note that a non-anticipatory policy does not know the positions of the long jobs. Thus, such a policy cannot start any of the stochastic jobs coming after the $(k \cdot m)$th long ones before time $k \cdot l$ for $1 \le k \le \lambda$. Recall that Z_{km} gives the position of the $(k \cdot m)$th long job. Hence, $n_s - Z_{km}$ stochastic jobs are delayed by $k \cdot l$.

For $\lambda = 1$, Lemma 7 *(iii)* implies that scheduling only J_s costs at least $l \frac{n_s}{4m}$, i.e.,

$$\sum_{j \in J_s} \mathbb{E}[C_j^0 \mid m \le X < 2m] \ge l \frac{n_s}{4m} \ge \frac{1}{4m}(\lambda+1) l n_d.$$

For $2 \le \lambda \le \lfloor \frac{n_s}{m} \rfloor$ let $\mathcal{E}$ denote the event that $\lambda m \le X < (\lambda+1)m$. With Lemma 7 *(ii)* it follows

$$\sum_{j \in J_s} \mathbb{E}[C_j^0 \mid \mathcal{E}] \ge \sum_{k=1}^{\lambda} l \mathbb{E}[\, n_s - Z_{km} \mid \mathcal{E}\,]$$
$$\ge l n_s \sum_{k=1}^{\lambda} \frac{\lambda m - km}{2\lambda m}$$
$$\ge \frac{l \lambda n_d}{8}$$
$$\ge \frac{1}{8}(\lambda+1) l n_d.$$

Using again the law of total expectation, we combine the above results for the different values of X and obtain

$$\sum_{j \in J} \mathbb{E}[C_j] \le \max\{8, 4m\} \sum_{j \in J} \mathbb{E}[C_j^*].$$ □

We conclude with a policy for scheduling Bernoulli-type instances of the scheduling problem $\mathrm{P}||\mathbb{E}[\sum C_j]$.

Algorithm 1: List scheduling policy for Bernoulli-type instances

At any time schedule as many available jobs as there are machines available in the following priority order:
if $m = 1$
  use SEPT
else if $n_d < m$ **and** $m \ge 2$
  use $J_s \prec J_d$
else if $n_s \le 2n_d$ **and** $n_d \ge m \ge 2$
  use $J_d \prec J_s$
else
  use $J_s \prec J_d$
end

Proof of Theorem 3. Algorithm 1 is a list scheduling policy that selects one out of three index policies, SEPT, $J_d \prec J_s$, and $J_s \prec J_d$, depending on the numbers of jobs and machines. The approximation guarantee follows from the fact that SEPT is optimal on a single machine [23] as well as Lemmas 5, 6, and 8. □

4. Concluding remarks

In this note, we rule out distribution-independent approximation factors for minsum scheduling for simple index policies, including SEPT, LEPT, and LVF. This strong lower bound holds even for Bernoulli-type instances. It may surprise that such most simple, yet stochastic, instances already seem to capture the inherent difficulties of stochastic scheduling. We believe that understanding the seemingly most simple Bernoulli-type instances is a key for making progress on approximative policies for stochastic scheduling problems.The general importance of high-variance jobs has also been observed in earlier work [19, 17, 24, 13, 12].

For Bernoulli-type instances we also give an $\mathcal{O}(m)$-approximative list scheduling policy. This result can be easily generalized to instances with arbitrary deterministic jobs when replacing (1) by a well-known lower bound [4]. The key ingredient to this analysis is the improved lower bound on the optimal cost due to exploiting the properties of the underlying probability distributions. It would be a major improvement to generalize this lower bound to arbitrary probability distributions. Generally, it is a common understanding that improving upon lower bounds is fundamental for designing $\mathcal{O}(1)$-approximative scheduling policies.

The setting with a fixed number of machines, m, is of particular interest. While the special case $m = 1$ is solved optimally by SEPT [23], even the problem on $m = 2$ machines is wide open. For simple Bernoulli-type instances, the index policy we give in this note is, in fact, a constant factor approximation. Any generalization would be of interest. Notice that our lower bound for arbitrary index policies as well as earlier lower bounds on SEPT [13, 2] rely on a large number of machines. Thus, even SEPT or some other simple index-policy might give a constant factor approximation for constant or bounded m.

For general instances, our lower bound for index policies suggests that future research shall investigate more sophisticated scheduling policies.

Acknowledgments. This research was supported by the German Science Foundation (DFG), ME 3825/1.

References

[1] J. L. Bruno, P. J. Downey, and G. N. Frederickson. Sequencing tasks with exponential service times to minimize the expected flow time or makespan. *Journal of the ACM*, 28(1):100–113, 1981.

[2] M. Cheung, F. Fischer, J. Matuschke, and N. Megow. An $\Omega(\Delta^{1/2})$ Gap example on the (W)SEPT Policy. Unpublished note, 2014.

[3] R. Conway, W. Maxwell, and L. Miller. *Theory of Scheduling*. Adison-Wesley Pub. Co., 1967.

[4] W. L. Eastman, S. Even, and I. M. Isaacs. Bounds for the optimal scheduling of n jobs on m processors. *Management Science*, 11(2):268–279, 1964.

[5] J. C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B*, 41:148–177, 1979.

[6] J. C. Gittins. *Multi-Armed Bandit Allocation Indices*. Wiley, New York, 1989.

[7] J. C. Gittins, K. D. Glazebrook, and R. R. Weber. *Multi-armed bandit allocation indices*. Wiley-Blackwell, London, 2nd edition, 2011.

[8] K. Glazebrook. Scheduling tasks with exponential service times on parallel processors. *Journal of Applied Probability*, 16(3):65–689, 1979.

[9] K. D. Glazebrook, D. J. Hodge, C. Kirkbride, and R. J. Minty. Stochastic scheduling: A short history of index policies and new approaches to index generation for dynamic resource allocation. *Journal of Scheduling*, 17(5):407–425, 2014.

[10] R. Graham. Bounds on multiprocessing timing anomalies. *SIAM Journal on Applied Mathematics*, 17(2):416–429, 1969.

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

[12] V. Gupta, B. Moseley, M. Uetz, and Q. Xie. Stochastic online scheduling on unrelated machines. In *IPCO*, volume 10328 of *Lecture Notes in Computer Science*, pages 228–240. Springer, 2017.

[13] S. Im, B. Moseley, and K. Pruhs. Stochastic scheduling of heavy-tailed jobs. In *STACS*, volume 30 of *LIPIcs*, pages 474–486. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[14] S. Jäger and M. Skutella. Generalizing the Kawaguchi-Kyan Bound to Stochastic Parallel Machine Scheduling. In *STACS*, volume 96 of *LIPIcs*, pages 43:1–43:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[15] B. Labonté. Ein Simulationssystem für stochastische Scheduling-Probleme und empirische Untersuchung zur Approximationsgüte von Politiken. Master's thesis, Technische Universität Berlin, 2013.

[16] C. McDiarmid. Concentration. In M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed, editors, *Probabilistic Methods for Algorithmic Discrete Mathematics*, pages 195–248, Berlin, 1998. Springer.

[17] N. Megow, M. Uetz, and T. Vredeveld. Models and algorithms for stochastic online scheduling. *Math. Oper. Res.*, 31(3):513–525, 2006.

[18] N. Megow and T. Vredefeld. A tight 2-approximation or preemptive stochastic scheduling. *Math. Oper. Res.*, 39(4):1297–1310, 2014.

[19] R. Möhring, A. Schulz, and M. Uetz. Approximation in stochastic scheduling: The power of LP-based priority policies. *Journal of the ACM*, 46(6):924–942, 1999.

[20] R. H. Möhring, F. J. Radermacher, and G. Weiss. Stochastic scheduling problems I – general strategies. *Zeitschrift für Operations Research*, 28(7):193–260, 1984.

[21] R. H. Möhring, F. J. Radermacher, and G. Weiss. Stochastic scheduling problems II – set strategies. *Zeitschrift für Operations Research*, 29:65–104, 1985.

[22] M. Pinedo and G. Weiss. The "largest variance first" policy in some stochastic scheduling problems. *Operations Research*, 35(6):884–891, 1987.

[23] M. H. Rothkopf. Scheduling with random service times. *Management Science*, 12(9):707–713, 1966.

[24] A. S. Schulz. Stochastic online scheduling revisited. In *Proceedings of COCOA*, volume 5165 of *Lecture Notes in Computer Science*, pages 448–457, Berlin, 2008. Springer.

[25] M. Skutella, M. Sviridenko, and M. Uetz. Unrelated machine scheduling with stochastic processing times. *Math. Oper. Res.*, 41(3):851–864, 2016.

[26] M. Skutella and M. Uetz. Stochastic machine scheduling with precedence constraints. *SIAM J. Comput.*, 34(4):788–802, 2005.

[27] W. E. Smith. Various optimizers for single–stage production. *Naval Research Logistics Quarterly*, 3(1-2):59–66, 6 1956.

[28] J. Walrand. *An Introduction to Queueing Networks*. Prentice Hall, Englewood Cliffs, NJ, 1988.

[29] 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.

[30] R. R. Weber. Scheduling jobs by stochastic processing requirements on parallel machines to minimize makespan or flowtime. *Journal of Applied Probability*, 19(1):167–182, 1982.

Appendix

In this section we prove the technical result about properties of Z_i (Lemma 7). Recall that Z_i is the random variable that denotes the position of the ith long job in a fixed sequence of the stochastic jobs J_s. We are not aware of any references regarding the distribution of Z_i conditioned on X, the number of long stochastic jobs. Thus, we give the prove here for the sake of completeness.

Lemma 7. *For X and Z_i defined as before and $1 \le i \le \lambda m, k \le n_s$ for $\lambda \in \{1, \dots, \lfloor \frac{n_s}{m} \rfloor\}$, the following holds:*

(i) $\mathbb{E}[Z_i \mid X = k] = \frac{i}{k+1}(n_s + 1)$.

(ii) $\mathbb{E}[Z_i \mid \lambda m \le X < (\lambda+1)m] \le \frac{i}{\lambda m+1}(n_s + 1)$.

(iii) $\mathbb{E}[n_s - \Pi_m \mid m \le X < 2m] \ge \frac{n_s}{4m}$.

Proof. We fix $1 \le r \le n_s$. Then, $X^{(r)} := \sum_{j=1}^{r} X_j$ follows a Binomial distribution with size parameter r and success probability $1/l$ as X_j are independent Bernoulli-distributed random variables with success probability $1/l$.

Let us recall that $\mathbb{P}[\mathcal{E} \mid \mathcal{F}] := \mathbb{P}[\mathcal{E} \cap \mathcal{F}]/\mathbb{P}[\mathcal{F}]$ for two events $\mathcal{E}$ and $\mathcal{F}$ with $\mathbb{P}[\mathcal{F}] > 0$.

ad (i). Let $1 \le i \le z \le k$. Then,

$$\{Z_i = z\} = \{X_z = 1\} \cap \{X^{(z-1)} = i - 1\},$$

i.e., the event that the ith long job is job z is equivalent to observing that the zth job is long after having seen $i-1$ long jobs among the stochastic jobs $1, \dots, z-1$.

Intersecting with the event $\{X = k\}$, we obtain

$$\begin{aligned}&\{X_z = 1\} \cap \{X^{(z-1)} = i-1\} \cap \{X = k\} = \\ &\quad \{X_z = 1\} \cap \{X^{(z-1)} = i-1\} \cap \{X - X^{(z)} = k - i\}.\end{aligned}$$

As the three events in the last line are independent of each other, we conclude

$$\begin{aligned}&\mathbb{P}[Z_i = z \mid X = k] = \\ &\frac{\mathbb{P}[X_z = 1] \cdot \mathbb{P}\big[X^{(z-1)} = i-1\big] \cdot \mathbb{P}\big[X - X^{(z)} = k-i\big]}{\mathbb{P}[X = k]} = \\ &\qquad \frac{\binom{z-1}{i-1} \cdot \binom{n_s - z}{k-i}}{\binom{n_s}{k}},\end{aligned}$$

where we used that $X^{(z-1)}$ and $X - X^{(z)}$ are Binomially distributed with success probability $1/l$ and size parameter $z-1$ and $n_s - z$, respectively.

With the convention $\binom{r}{q} = 0$ for $r, q \in \mathbb{N}$ with $q > r$, it follows

$$\begin{aligned}\binom{n_s}{k}\mathbb{E}[\,Z_i \mid X = k\,] &= \sum_{z=0}^{n_s} z\,\mathbb{P}[Z_i = z \mid X = k]\binom{n_s}{k} \\ &= i\sum_{z=0}^{n_s}\binom{z}{i}\binom{n_s - z}{k-i} \\ &= i\binom{n_s+1}{k+1}.\end{aligned}$$

where the last equality follows from an index shift – $\sum_{z=0}^{n_s}\binom{z}{i}\binom{n_s-z}{k-i} = \sum_{z=1}^{n_s+1}\binom{z-1}{i}\binom{n_s+1-z}{k-i}$ – and the following observation: The last line in the above calculation asks in how many ways you can pick $k+1$ successes among n_s+1 trials. We can partition this based on the position of the $(i+1)$st success for a fixed i with a similar idea as used above. The $(i+1)$st success can be positioned between the $(i+1)$st and the $(n-k+i)$th trial. If the $(i+1)$st success is at position l, there have to be i successes among the first $l-1$ trials and, since we want to pick $k+1$ successes, the remaining $n_s + 1 - l$ trials have to contain $k-i$ successes. Summing over all positions l of the $i+1$st success, yields the equality.

ad (ii). With the law of total expectation, we can use *(i)* to prove the statement as follows. Indeed, conditioning on the event $X = k$ for $\lambda m \le k < (\lambda+1)m$ yields

$$\begin{aligned}&\mathbb{E}[\,Z_i \mid \lambda m \le X < (\lambda+1)m\,] \\ &= \sum_{k=\lambda m}^{(\lambda+1)m-1} \mathbb{E}[\,Z_i \mid X = k\,]\mathbb{P}[X = k \mid \lambda m \le X < (\lambda+1)m].\end{aligned}$$

Applying *(i)*, we get

$$= \sum_{k=\lambda m}^{(\lambda+1)m-1} \frac{i}{k+1}(n_s+1)\mathbb{P}[X = k \mid \lambda m \le X < (\lambda+1)m].$$

As $k = \lambda m$ clearly is an upper bound on every summand, this yields

$$\le \sum_{k=\lambda m}^{(\lambda+1)m-1} \frac{i}{\lambda m+1}(n_s+1)\mathbb{P}[X = k \mid \lambda m \le X < (\lambda+1)m].$$

The law of total expectation concludes the calculation:

$$= \frac{i}{\lambda m+1}(n_s+1).$$

ad (iii). With *(i)* it follows that

$$\begin{aligned}\mathbb{E}[\,n_s - \Pi_m \mid X = m\,] &= n_s - \frac{m}{m+1}(n_s+1) \\ &= \frac{n_s m + n_s - n_s m - m}{m+1} \\ &\ge \frac{n_s}{4m},\end{aligned}$$

where we used $n_s > 2m$ for the last inequality. Using again the law of total statement as in *(ii)*, the statement follows. □