Learning in Stochastic Machine Scheduling

Sebastián Marbán, Cyriel Rutten, and Tjark Vredeveld

Department of Quantitative Economics, Maastricht University
P.O. Box 616, 6200 MD Maastricht, The Netherlands
{s.marban,c.rutten,t.vredeveld}@maastrichtuniversity.nl

Abstract. We consider a scheduling problem in which two classes of independent jobs have to be processed non-preemptively by a single machine. The processing times of the jobs are assumed to be exponentially distributed with parameters depending on the class of each job. The objective is to minimize the sum of expected completion times. We adopt a Bayesian framework in which both job class parameters are assumed to be unknown. However, by processing jobs from the corresponding class, the scheduler can gradually learn about the value of these parameters, thereby enhancing the decision making in the future.

For the traditional stochastic scheduling variant, in which the parameters are known, the policy that always processes a job with Shortest Expected Processing Time (SEPT) is an optimal policy. In this paper, we show that in the Bayesian framework the performance of SEPT is at most a factor 2 away from the performance of an optimal policy. Furthermore, we introduce a second policy learning-SEPT (ℓ-SEPT), which is an adaptive variant of SEPT. We show that ℓ-SEPT is no worse than SEPT and empirically outperforms SEPT. However, both policies have the same worst-case performance, that is, the bound of 2 is tight for both policies.

1 Introduction

In this paper, we consider the classical non-preemptive single machine scheduling problem to minimize the total completion time. In deterministic and traditional stochastic scheduling, this problem is well understood and can be solved to optimality by the Shortest (Expected) Processing Time (SPT or SEPT) policy: process the jobs in non-decreasing order of their (expected) processing time [19,22]. In traditional stochastic scheduling, it is assumed that the jobs' processing times are independent random variables of which the parameters, such as the expected value, are fully known. We relax this assumption by introducing parameter uncertainty. Like in [2,8,10,11,12], we adopt a Bayesian viewpoint in which we have prior distributions for the uncertain parameters. These priors represent our beliefs on the values of the parameters. Furthermore, the Bayesian framework allows us to learn about the value of the parameters by processing jobs and observing their realized processing times. However, experimenting with different jobs to learn about the value of the corresponding parameters can be costly in terms of the waiting times of the still to be processed jobs. Hence, learning

R. Solis-Oba and G. Persiano (Eds.): WAOA 2011, LNCS 7164, pp. 21–34, 2012.

should be conducted carefully in order to minimize the sum of completion times in expectation.

Problem definition. There are two classes of independent jobs that have to be processed by a single machine. Each class J_i consists of n_i jobs ($i = 1, 2$). All jobs are available for processing from the beginning and preemption of jobs is not allowed, that is, once a job has been initiated it must remain on the machine until completion. The processing time of a job in class J_i is a random variable, which is independently and exponentially distributed with parameter ϑ_i. Distinguishing from traditional stochastic scheduling, in the scheduling model under consideration the value of ϑ_i is unknown. The goal is to minimize the total completion time in expectation, $\sum_j \mathbf{E}[C_j]$.

We introduce a random variable Θ_i describing the scheduler's beliefs regarding the value of ϑ_i. In the Bayesian approach, ϑ_i can be considered as a realization of the random variable Θ_i. The initial distribution of Θ_i, that is, before any job has been processed, is called the prior. As in [11,12], we assume that the prior is a gamma distribution with parameters $\omega_i > 0$ and $\alpha_i > 1$. Depending on the confidence in his beliefs about ϑ_i, the scheduler can choose the values of ω_i and α_i such that the prior is very peaked (the scheduler is very certain about his beliefs) or relatively flat (the scheduler is not certain about his beliefs) or anywhere in between.

After a job of class J_i is processed, we observe this job's processing time x. Since the gamma distribution is a conjugate prior for the exponential distribution, the posterior distribution of Θ_i, representing the beliefs of ϑ_i after having observed processing time realization x, is a gamma distribution with parameters $\omega_i + x$ and $\alpha_i + 1$. This result is stated in a.o. Section 9.4 of [5] and is also derived from Bayes' theorem for probability density functions. In this way, the scheduler gradually learns about the unknown parameter, thereby enhancing his decision making in the future.

A solution to a stochastic scheduling problem is not merely a simple schedule, but a so-called *scheduling policy*. We follow the notion of scheduling policies as proposed by Möhring, Radermacher, and Weiss [17]. A scheduling policy makes decisions on which job to schedule at certain decision times. We require a policy to be *non-anticipatory*: at any time, it may not utilize the actual processing times of jobs that have not yet been completed. A scheduling policy may, of course, at any decision time, use the information that it has gathered up to this time. An *optimal scheduling policy* is defined as a non-anticipatory scheduling policy that minimizes the objective value in expectation. Note hereby that an optimal scheduling policy underlies the uncertainty about processing times as well as the uncertainty about the parameters.

Burnetas and Katehakis [2] and Hamada and Glazebrook [11] present optimal policies for different number of job classes. Even for the case of two job classes, one of which has known parameter, these policies require solving extensive dynamic programs. This is in contrast to the traditional stochastic scheduling variant of the problem in which the optimal scheduling policy is SEPT [19]. The reason why SEPT is not an optimal policy in the Bayesian setting lies in the fact that when the expected processing times of the job classes are close to each other and the parameter of the class with higher expected value is more uncertain it may be beneficial to learn about the value of the underlying parameter of this class. As SEPT is a very simple policy that is optimal for the traditional stochastic scheduling problem under consideration, it is interesting to know how well it performs in the setting with parameter uncertainty.

In the Bayesian setting, there are two natural versions of SEPT. The first one, which we keep calling SEPT, determines the order in which the jobs will be processed at the beginning based on its initial beliefs. The second version, which we denote by *learning*-SEPT or ℓ-SEPT, updates its beliefs on ϑ_i every time a job of class J_i is completed. After each completion of a job, ℓ-SEPT will schedule the job with shortest expected processing time with respect to its current beliefs. In this paper, we investigate the quality of the solution value obtained by both policies. Adopting the definition of [18], we define a policy Π to be a *ρ-approximative policy* when $\mathbf{E}\left[\Pi(I)\right] \leq \rho\,\mathbf{E}\left[\mathrm{OPT}(I)\right]$ on any scheduling instance I. Here $\mathbf{E}\left[\Pi(I)\right]$ is the expected total completion time of policy Π on instance I and OPT is the optimal non-anticipatory policy. The value ρ is called the *(worst case) performance guarantee.*

Related work. In traditional stochastic scheduling, the processing times of jobs are random variables for which the parameters of the underlying distribution are known. Rothkopf [19] shows that WSEPT (Weighted Shortest Expected Processing Time) is an optimal policy for the stochastic single machine scheduling problem, where the objective is to minimize the sum of weighted expected completion times. Weiss [23,24] analyzes the performance of WSEPT for the stochastic parallel machine scheduling problem. He shows asymptotic optimality of WSEPT for a certain class of processing time distributions. The first guarantee on the quality of an approximative policy was given by Möhring, Schulz, and Uetz [18]. Other approximative policies have been considered in [4,15,16,21].

This paper contributes to the field by applying a Bayesian framework to the single machine scheduling problem. Examples of papers that apply the same framework to scheduling problems are limited. In the pioneering paper of Gittins and Glazebrook [8], the distributions of processing times of jobs depend all upon the same unknown parameter. The optimal schedule is obtained by calculating appropriate dynamic allocation indices, first proposed by Gittins and Jones [9]. Hamada and Glazebrook [11] present another example studying the Bayesian scheduling problem with multiple weighted job classes. Optimal policies are derived using dynamic allocation indices similar to the ones in [7]. Burnetas and Katehakis [2] derive optimality conditions for the same problem with two job classes: one with known and one with unknown underlying parameter. Glazebrook and Owen [10] quantify the difference between adaptive scheduling policies based on Bayesian methodology and non-adaptive classical stochastic scheduling policies.

Bayesian methodology is widely applied in research fields related to scheduling. In inventory management for example, there is a large body of literature dealing with uncertain demand distributions and Bayesian learning. Pioneered by [20], some recent papers are given by [3,13]. The majority of these papers assumes that prices are exogenous and studies the problem of making optimal inventory decisions. Bayesian demand learning has also received a great deal of attention within the field of pricing, see [1,14]. All these papers are experimental in that they focus on developing heuristics and studying their computational aspects. The first, and so far only, paper to analyze the theoretical worst-case performance of a Bayesian pricing heuristic is [6].

Our results. In Section 3, we first show that ℓ-SEPT is in expectation better than the non-adaptive version SEPT. Furthermore, we show that the performance guarantee for both SEPT and ℓ-SEPT is a function depending on the number of jobs in both classes and that this function can be arbitrarily close to, but is bounded by, 2. If one of the two job classes has a constant number of jobs and the number of jobs of the other class tends to infinity, then SEPT and linebreak ℓ-SEPT are asymptotically optimal. In Section 4, we show that the bound for SEPT as well as the bound for ℓ-SEPT is tight. To the best of our knowledge, this is one of the first tight performance guarantees in stochastic scheduling, where the tightness follows from non-degenerate processing time distributions. Section 5 complements our theoretical findings with some preliminary computational results, showing that ℓ-SEPT in practice outperforms the non-adaptive variant, although the worst-case performance guarantees are the same. Finally, we conclude with some remarks on the case of m job classes.

2 Preliminaries and Scheduling Policies

In this section, we introduce the Bayesian scheduling framework and policies SEPT, ℓ-SEPT, and OPT. Additionally, we give useful bounds on the performance of these policies.

2.1 Bayesian Methodology

Bayesian methodology offers a method to formally recognize the uncertainty regarding parameter ϑ_i. A random variable Θ_i is introduced which describes the scheduler's beliefs regarding the value of ϑ_i. In the Bayesian approach, ϑ_i can be considered as a realization of the random variable Θ_i. For some $\theta > 0$, let $g_i(\theta) := \frac{\partial}{\partial\theta}\mathbf{Pr}[\Theta_i \leq \theta]$ denote a (prior) probability density function. Intuitively, the probability expresses how strongly we believe that the value of ϑ_i is less than or equal to θ, prior to seeing any realization of processing times of jobs of class J_i. We assume $g_i(\theta)$ to be a gamma distribution with parameters $\omega_i > 0$ and $\alpha_i > 1$. Once k jobs of class J_i have been completed with processing time realizations x_1 up to x_k, the beliefs with respect to the unknown value of ϑ_i will be updated and expressed by the (posterior) probability density function

$$g_i(\theta|x_1,\ldots,x_k) := \frac{\partial}{\partial\theta}\mathbf{Pr}[\Theta_i \leq \theta | X_1 = x_1,\ldots,X_k = x_k] \,.$$

Since the gamma distribution provides a conjugate prior for the exponential distribution, the posterior $g_i(\theta|x_1,\ldots,x_k)$ is also a gamma distribution with parameters $\omega_i' := \omega_i + \sum_{j=1}^{k} x_j$ and $\alpha_i' := \alpha_i + k$ (see e.g. Section 9.4 of [5]).

Updating beliefs toward ϑ_i results in updated beliefs regarding the processing times of uncompleted jobs in class J_i. The probability density function expressing these latter beliefs, after having completed k jobs of class J_i, is denoted by

$$f_{i,k+1}(x_{k+1}) := \frac{\partial}{\partial x_{k+1}} \mathbf{Pr}\left[X_{k+1} \leq x_{k+1} | X_1 = x_1, \ldots, X_k = x_k\right] ,$$

which is equal to

$$\begin{aligned} f_{i,k+1}(x_{k+1}) &= \int_0^\infty f(x_{k+1}|\theta) g_i(\theta|x_1, \ldots, x_k) \partial\theta \qquad (1) \\ &= \int_0^\infty \theta e^{-\theta x_{k+1}} \frac{{\omega'_i}^{\alpha'_i}}{\Gamma(\alpha'_i)} \theta^{\alpha'_i - 1} e^{-\theta \omega'_i} \partial\theta = \frac{\alpha'_i {\omega'_i}^{\alpha'_i}}{(\omega'_i + x_{k+1})^{\alpha'_i + 1}}, \end{aligned}$$

where $f(x_{k+1}|\theta)$ is an exponential probability density function with parameter θ. Furthermore, straightforward integration yields the first moment of X_{k+1}:

$$\mathbf{E}\left[X_{k+1}|x_1, \ldots, x_k\right] = \int_0^\infty x_{k+1}\, f_{i,k+1}(x_{k+1}) \partial x_{k+1} = \frac{\omega_i + \sum_{j=1}^k x_j}{\alpha_i + k - 1}. \qquad (2)$$

The more jobs of job class i have been processed, the more accurate the scheduler's beliefs regarding ϑ_i will be. First, the expected value of $(\Theta_i|x_1, \ldots, x_k)$ will converge to ϑ_i by the law of large numbers. Secondly, the variance of $(\Theta_i|x_1, \ldots, x_k)$ will decrease since ω_i and α_i will be increased with every new observation. Hence, the more jobs we process, the more peaked and the more centered around ϑ_i the distribution of $(\Theta_i|x_1, \ldots, x_k)$ will become, i. e., the more we learn about the value of ϑ_i.

2.2 Bayesian Scheduling Policies

An optimal policy for the Bayesian scheduling problem at hand, OPT, minimizes total completion time in expectation, thereby taking into account the uncertainty regarding the job class parameters. That is, the values of the parameters ϑ_i are unknown to OPT, but the policy will anticipate and act in its decision making upon the additional information to be revealed when processing a job of a certain class. In order to characterize OPT, we formulate the problem as a dynamic program, introduced by [11].

Let $\boldsymbol{n} = (n_1, n_2)$, $\boldsymbol{\omega} = (\omega_1, \omega_2)$, and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$. Then, $(\boldsymbol{n}, \boldsymbol{\omega}, \boldsymbol{\alpha}) = (n_1, n_2, \boldsymbol{\omega}, \boldsymbol{\alpha}) \in \mathbb{Z}_+^2 \times \mathbb{R}_{>0}^2 \times \mathbb{R}_{>1}^2$ denotes a state vector encompassing all relevant information of the state the system is in. It consists of the number of jobs in each class J_i as well as the parameters of the current belief for ϑ_i. Let $\boldsymbol{e}_i$ be the ith unit vector. If in state $(\boldsymbol{n}, \boldsymbol{\omega}, \boldsymbol{\alpha})$, a job of class J_i is processed and completed having realization x, then the state changes to $(\boldsymbol{n} - \boldsymbol{e}_i, \boldsymbol{\omega} + x\boldsymbol{e}_i, \boldsymbol{\alpha} + \boldsymbol{e}_i)$. Let $\mathbf{E}\left[\Pi^*(\boldsymbol{n}, \boldsymbol{\omega}, \boldsymbol{\alpha})\right]$ denote the expected sum of completion times when the optimal policy is adopted from state $(\boldsymbol{n}, \boldsymbol{\omega}, \boldsymbol{\alpha})$ onwards. Further, let $\mathbf{E}\left[\Pi_i^*(\boldsymbol{n}, \boldsymbol{\omega}, \boldsymbol{\alpha})\right]$ denote the sum of the expected completion times of a policy which first processes a job of class J_i (assuming $n_i \geq 1$) and follows an optimal policy afterwards.

An optimal policy can then be modeled by the following dynamic program:

$$\mathbf{E}\left[\Pi^*(\boldsymbol{n},\boldsymbol{\omega},\boldsymbol{\alpha})\right] = \min\left\{\mathbf{E}\left[\Pi_1^*(\boldsymbol{n},\boldsymbol{\omega},\boldsymbol{\alpha})\right],\mathbf{E}\left[\Pi_2^*(\boldsymbol{n},\boldsymbol{\omega},\boldsymbol{\alpha})\right]\right\} \quad \forall\, \boldsymbol{n} \geq \boldsymbol{1} \tag{3}$$

and

$$\begin{aligned}\mathbf{E}\left[\Pi^*(n_1,0,\boldsymbol{\omega},\boldsymbol{\alpha})\right] &= \left(\sum_{i=1}^{n_1} i\right)\frac{\omega_1}{\alpha_1-1} = \frac{n_1(n_1+1)}{2}\frac{\omega_1}{\alpha_1-1} \qquad \forall\, n_1 \geq 0,\\ \mathbf{E}\left[\Pi^*(0,n_2,\boldsymbol{\omega},\boldsymbol{\alpha})\right] &= \left(\sum_{i=1}^{n_2} i\right)\frac{\omega_2}{\alpha_2-1} = \frac{n_2(n_2+1)}{2}\frac{\omega_2}{\alpha_2-1} \qquad \forall\, n_2 \geq 0.\end{aligned}$$

As the length of the first job to be processed by a policy influences the completion time of all jobs, straightforward calculations show that

$$\mathbf{E}\left[\Pi_i^*(\boldsymbol{n},\boldsymbol{\omega},\boldsymbol{\alpha})\right] = (n_1+n_2)\frac{\omega_i}{\alpha_i-1} + \int_0^\infty \mathbf{E}\left[\Pi^*(\boldsymbol{n}-\boldsymbol{e}_i,\boldsymbol{\omega}+x\boldsymbol{e}_i,\boldsymbol{\alpha}+\boldsymbol{e}_i)\right] f_{i1}(x)dx, \tag{4}$$

for all $n_i \geq 1$.

In the traditional stochastic scheduling variant, in which the parameters ϑ_i are known, the policy SEPT processes jobs in non-decreasing order of expected processing times. In the Bayesian scheduling problem at hand, SEPT processes the jobs of each job class en bloc, starting with the class having the shortest expected processing time. Formally, SEPT starts processing all jobs of class J_1 in case $\frac{\omega_1}{\alpha_1-1} < \frac{\omega_1}{\alpha_1-1}$ followed by all jobs of class J_2, and vice versa otherwise. The random variable for the sum of completion times of SEPT is denoted by Π^s, and its expected value can be written as

$$\begin{aligned}\mathbf{E}\left[\Pi^s(n_1,n_2,\boldsymbol{\omega},\boldsymbol{\alpha})\right] =& \frac{n_1(n_1+1)}{2}\frac{\omega_1}{\alpha_1-1} + \frac{n_2(n_2+1)}{2}\frac{\omega_2}{\alpha_2-1}\\ &+ n_1 n_2 \min\left\{\frac{\omega_1}{\alpha_1-1},\frac{\omega_2}{\alpha_2-1}\right\}.\end{aligned} \tag{5}$$

The non-adaptive character of SEPT could result in performance loss in comparison to a policy which makes use of additional information being revealed when processing the jobs. This shortcoming of SEPT is illustrated by the following example.

Example 1. Consider the Bayesian scheduling problem with two job classes. Let $\omega_1 = 10, \alpha_1 - 1 = 90, \omega_2 = 0.2$ and $\alpha_2 - 1 = 2$ such that $\mathbf{E}\left[X_1\right] = \frac{\omega_1}{\alpha_1-1} = \frac{10}{90} > 0.1$ and $\mathbf{E}\left[X_2\right] = \frac{\omega_2}{\alpha_2-1} = \frac{0.2}{2} = 0.1$, where X_i denotes the processing time of the first job to be processed of class J_i. Since $\mathbf{E}\left[X_1\right] > \mathbf{E}\left[X_2\right]$, SEPT will first process all jobs of class J_2 and afterward all jobs of class J_1. However, we picked our values in such a way that the distribution of Θ_1 is peaked, i. e., we are relatively sure about the value of ϑ_1, whereas the distribution of Θ_2 is flat, i. e., we are relatively unsure about the value of ϑ_2 (see Figure 1). Consequently, it might be that actually $\vartheta_2 < \vartheta_1$, such that, in contrast to SEPT, it would be best to first start processing all jobs of class J_1. Just like SEPT, OPT will start processing the jobs of class J_2 since $\mathbf{E}[X_2] < \mathbf{E}[X_1]$ and the beliefs regarding ϑ_2 are not that strong. However, in case $\vartheta_2 < \vartheta_1$, OPT will observe high processing times for the first few jobs of job class J_2 and realize his mistake. After processing a few jobs of job class J_2, OPT will therefore switch to processing jobs of class J_1, whereas SEPT continues with processing all jobs of job class J_2. By choosing appropriate values for the parameters $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$ the probability that $\vartheta_2 < \vartheta_1$ can be made even larger. Hence, the performance of SEPT can be far away from that of OPT.

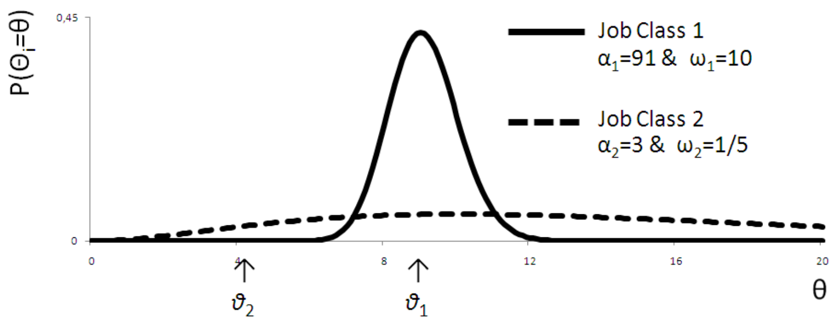

Fig. 1. Gamma distributions describing the beliefs with respect to the unknown parameters ϑ_1 and ϑ_2. Since the distribution corresponding to job class J_1 (J_2) is relatively peaked (flat), we are quite sure (unsure) about the value of ϑ_1 (ϑ_2).

To overcome the shortcoming discussed in the example above, we propose an adaptive policy *learning*-SEPT (ℓ-SEPT). Whenever the machine is idle, this policy starts processing the job with shortest expected processing time. Thereby, it updates the expected processing time of jobs in a class every time a job of this specific class has been completed. Formally, after k_1 jobs of class J_1 and k_2 jobs of class J_2 have been finished, ℓ-SEPT starts processing a job of class J_1 in case $\frac{\omega_1+\sum_{j=1}^{k_1} x_j}{\alpha_1+k_1-1} < \frac{\omega_2+\sum_{j=1}^{k_2} y_j}{\alpha_2+k_2-1}$, and a job of class J_2 otherwise, where x_i denotes the observed value of the processing time of the ith job of class J_1 and y_j denotes the realized value of the processing time of the jth job of class J_2. Note that in Example 1, ℓ-SEPT also starts processing jobs of class J_2. However, in case $\vartheta_2 < \vartheta_1$, just like OPT, ℓ-SEPT will realize his mistake after having processed a few jobs of class J_2 and continue with processing jobs of class J_1. In what follows, Π^ℓ denotes the random variable for the sum of completion times when policy ℓ-SEPT is used.

To summarize, we observe that ℓ-SEPT uses more information than SEPT whereas OPT uses all available information, although none of the three policies know the values of ϑ_i. All three policies know the values of ω_i and α_i which are derived from the scheduler's beliefs about ϑ_i. Based on these values SEPT processes first all jobs of the job class with minimal expected processing time for the first job to be processed. OPT and ℓ-SEPT are more intelligent in the sense that they make use of the underlying distribution of Θ_i and update this distribution in light of new realizations. OPT in particular uses $g_i(\theta|x_1,\ldots,x_k)$ through equations (1), (3), and (4). ℓ-SEPT actually only uses the first moment of the updated distribution of $(\Theta_i|x_1,\ldots,x_k)$ to determine that the expected processing time of the next job of job class J_i equals (2), once k jobs of job class J_i have been processed.

In terms of decision making, one could thus interpret OPT as having a long-term view whereas SEPT and ℓ-SEPT both have a short-term view. Both policies process a job of class J_i only if the expected processing time of the next job in this class is minimal. OPT, however, might choose to process a job of class J_i for which the expected processing time is not necessarily minimal. As a trade-off, OPT benefits from the additional information which is acquired regarding the uncertain parameter ϑ_i. This information could then lead to better future decision making and a lower sum of completion times.

2.3 Bounds on Scheduling Policies

A trivial lower bound on the performance of an arbitrary policy is based on the fact that in any policy jobs of a class have to wait for other jobs of the same class. Hence, in constructing the lower bound we neglect waiting times caused by jobs having to wait for jobs of a different class.

Lemma 1. *Let Π be an arbitrary scheduling policy. Then, for any $n_1, n_2 \geq \mathbf{0}$, $\boldsymbol{\omega} > \mathbf{0}$, and $\boldsymbol{\alpha} > \mathbf{1}$,*

$$\begin{aligned}\mathbf{E}\left[\Pi(n_1,n_2,\boldsymbol{\omega},\boldsymbol{\alpha})\right] &\geq \mathbf{E}\left[\Pi(n_1,0,\boldsymbol{\omega},\boldsymbol{\alpha})\right] + \mathbf{E}\left[\Pi(0,n_2,\boldsymbol{\omega},\boldsymbol{\alpha})\right] \\ &= \frac{(n_1+1)n_1}{2}\frac{\omega_1}{\alpha_1-1} + \frac{(n_2+1)n_2}{2}\frac{\omega_2}{\alpha_2-1}.\end{aligned}$$

As the expected completion time of each job is delayed by the expected processing time of the first job to be processed by the optimal policy, we can bound the value of the optimal policy as in the following lemma.

Lemma 2. *For any $n_1, n_2 \geq \mathbf{0}$, $\boldsymbol{\omega} > \mathbf{0}$, and $\boldsymbol{\alpha} > \mathbf{1}$,*

$$\begin{aligned}\mathbf{E}\left[\Pi^*(n_1,n_2,\boldsymbol{\omega},\boldsymbol{\alpha})\right] \geq\ & \frac{n_1(n_1+1)}{2}\frac{\omega_1}{\alpha_1-1} + \frac{n_2(n_2+1)}{2}\frac{\omega_2}{\alpha_2-1} \\ & + \min\{n_1,n_2\}\min\left\{\frac{\omega_1}{\alpha_1-1},\frac{\omega_2}{\alpha_2-1}\right\}.\end{aligned}$$

3 Upper Bound on Performance Guarantees

In this section, we prove that both SEPT and ℓ-SEPT have a performance guarantee less than 2. First, we show that the adaptive policy is indeed better than sequencing the jobs a priori. The proof of this theorem is postponed to the full version.

Theorem 1. *For any $\boldsymbol{n} \geq \mathbf{0}$, $\boldsymbol{\omega} > \mathbf{0}$, and $\boldsymbol{\alpha} > \mathbf{1}$,*

$$\mathbf{E}\left[\Pi^{\ell}(\boldsymbol{n},\boldsymbol{\omega},\boldsymbol{\alpha})\right] \leq \mathbf{E}\left[\Pi^{s}(\boldsymbol{n},\boldsymbol{\omega},\boldsymbol{\alpha})\right].$$

Given the relation between SEPT and ℓ-SEPT, we can prove the performance guarantee on both SEPT and ℓ-SEPT.

Theorem 2. *For any $n_1, n_2 \geq 0$, $\boldsymbol{\omega} > \mathbf{0}$, and $\boldsymbol{\alpha} > \mathbf{1}$,*

$$\frac{\mathbf{E}\left[\Pi^{s}(n_1,n_2,\boldsymbol{\omega},\boldsymbol{\alpha})\right]}{\mathbf{E}\left[\Pi^{*}(n_1,n_2,\boldsymbol{\omega},\boldsymbol{\alpha})\right]} \leq \frac{n_1^2+n_2^2+2n_1n_2+n_1+n_2}{n_1^2+n_2^2+n_1+n_2+2\min\{n_1,n_2\}} < 2.$$

Proof. The first inequality follows directly from Theorem 1. To prove the second and last inequality, let $n_1, n_2 \geq 0$, $\boldsymbol{\omega} > \mathbf{0}$, and $\boldsymbol{\alpha} > \mathbf{1}$. Combining (5) and Lemma 2, we obtain

$$\frac{\mathbf{E}\left[\Pi^{s}(n_1,n_2,\boldsymbol{\omega},\boldsymbol{\alpha})\right]}{\mathbf{E}\left[\Pi^{*}(n_1,n_2,\boldsymbol{\omega},\boldsymbol{\alpha})\right]} \leq \frac{n_1(n_1+1)\frac{\omega_1}{\alpha_1-1}+n_2(n_2+1)\frac{\omega_2}{\alpha_2-1}+2n_1n_2\min\left\{\frac{\omega_1}{\alpha_1-1},\frac{\omega_2}{\alpha_2-1}\right\}}{n_1(n_1+1)\frac{\omega_1}{\alpha_1-1}+n_2(n_2+1)\frac{\omega_2}{\alpha_2-1}+2\min\{n_1,n_2\}\min\left\{\frac{\omega_1}{\alpha_1-1},\frac{\omega_2}{\alpha_2-1}\right\}}$$

observing that for any $0 < c \leq b$ and $0 < d \leq a$, it holds that $\frac{a+b}{a+c} \leq \frac{d+b}{d+c}$ and replacing $\frac{\omega_1}{\alpha_1-1}$ and $\frac{\omega_2}{\alpha_2-1}$ by the minimum of the two, we can bound this by

$$\leq \frac{n_1^2+n_2^2+2n_1n_2+n_1+n_2}{n_1^2+n_2^2+n_1+n_2+2\min\{n_1,n_2\}} \leq \frac{4n_{\max}^2+2n_{\max}}{2n_{\max}^2+2n_{\max}} < 2,$$

where $n_{\max} = \max\{n_1,n_2\}$.

Note that it follows from Theorem 2 that the performance guarantee will be close to one in case the number of jobs in one class is of a different order than the number of jobs in the second class. To be more explicit, when the number of jobs in one class is fixed while the number of jobs in the second class tends to infinity, then the performance guarantee will go to one, yielding asymptotic optimality of SEPT and ℓ-SEPT.

4 Tightness of the Performance Guarantees

In this section, we show that the performance guarantee shown in the previous section is tight for SEPT as well as ℓ-SEPT. Although by Theorem 1, it suffices to show that the guarantee of ℓ-SEPT is tight, we first give a lower bound on the performance guarantee of SEPT, as this one is more intuitive, whereas the lower bound for ℓ-SEPT is rather technical.

4.1 Lower Bound on the Performance Guarantee of SEPT

We show that for any $\epsilon > 0$ there exists an instance for which the ratio of the value of SEPT to the value of OPT is only an additive ϵ away from the performance guarantee of Theorem 2. In order to obtain this result, we make use of the following two facts.

Fact 1. *For any $\omega > 0$ and $\alpha > 1$,*

$$\int_0^\infty \min\left\{\frac{\omega + x}{\alpha}, 1\right\} f_{11}(x)dx = \frac{\omega}{\alpha - 1} - \frac{1}{\alpha - 1}\left(\frac{\omega}{\alpha}\right)^\alpha.$$

Fact 2. *For any $\alpha > 1$,*

$$\lim_{\alpha \downarrow 1} \frac{1}{\alpha - 1}\left(\frac{\alpha - 1}{\alpha}\right)^\alpha = 1$$

Additionally, we need a lower bound on SEPT and an upper bound on OPT.

Lemma 3. *For any $n_1, n_2 \geq 0$, there exist parameter settings $\boldsymbol{\omega} > \mathbf{0}$, and $\boldsymbol{\alpha} > \mathbf{1}$ such that $\frac{\omega_1}{\alpha_1 - 1} < \frac{\omega_2}{\alpha_2 - 1} = 1$ and*

$$\mathbf{E}\left[\Pi^s(n_1, n_2, \boldsymbol{\omega}, \boldsymbol{\alpha})\right] > \frac{n_1(n_1+1)}{2} + \frac{n_2(n_2+1)}{2} + n_1 n_2 - \epsilon,$$

for any $\epsilon > 0$.

Proof. For all $\epsilon' > 0$ and arbitrary $\alpha_1 > 1$, let $\omega_1 = (1 - \epsilon')(\alpha_1 - 1)$. By (5), we have

$$\mathbf{E}\left[\Pi^s(n_1, n_2, \boldsymbol{\omega}, \boldsymbol{\alpha})\right] = \frac{n_1(n_1+1)}{2}(1 - \epsilon') + \frac{n_2(n_2+1)}{2} + n_1 n_2(1 - \epsilon').$$

Hence, for any $\epsilon > 0$, there exists an $\epsilon' > 0$ for which the lemma holds.

Lemma 4. *For any $n_1, n_2 \geq 0$, there exist parameter settings $\boldsymbol{\omega} > \mathbf{0}$, and $\boldsymbol{\alpha} > \mathbf{1}$ such that $\frac{\omega_1}{\alpha_1 - 1} < \frac{\omega_2}{\alpha_2 - 1} = 1$ and*

$$\mathbf{E}\left[\Pi^*(n_1, n_2, \boldsymbol{\omega}, \boldsymbol{\alpha})\right] < n_1 + \frac{n_1(n_1+1)}{2} + \frac{n_2(n_2+1)}{2} + \epsilon,$$

for any $\epsilon > 0$.

Proof. Consider the following policy Π: first process one job of class J_2, observing realization y, and schedule all remaining jobs according to SEPT. That is, if $\frac{\omega_1}{\alpha_1 - 1} \leq \frac{\omega_2 + y}{\alpha_2}$ then process first all jobs of class J_1 and then the remaining jobs of class J_2 and otherwise first process the remaining jobs of class J_2 and then all jobs of class J_1. Using y to denote the observed value of the first job of class J_2, we have that for any $n_1, n_2 \geq 0$, $\boldsymbol{\omega} > \mathbf{0}$, and $\boldsymbol{\alpha} > \mathbf{1}$ such that $\frac{\omega_1}{\alpha_1 - 1} < \frac{\omega_2}{\alpha_2 - 1} = 1$,

$$\begin{aligned}
&\mathbf{E}\left[\Pi^*(n_1, n_2, \boldsymbol{\omega}, \boldsymbol{\alpha})\right] \leq \mathbf{E}\left[\Pi(n_1, n_2, \boldsymbol{\omega}, \boldsymbol{\alpha})\right] \\
&\quad = (n_1 + n_2)\frac{\omega_2}{\alpha_2 - 1} + \int_0^\infty \mathbf{E}\left[\Pi^s(n_1, n_2 - 1, \boldsymbol{\omega} + y\boldsymbol{e}_2, \boldsymbol{\alpha} + \boldsymbol{e}_2)\right] f_{21}(y)dy \\
&\quad \overset{(5)}{=} n_1 + n_2 + \frac{n_1(n_1+1)}{2}\frac{\omega_1}{\alpha_1 - 1} + \frac{n_2(n_2-1)}{2} + n_1(n_2 - 1)\int_0^\infty \min\left\{\frac{\omega_1}{\alpha_1 - 1}, \frac{\omega_2 + y}{\alpha_2}\right\} f_{21}(y)dy \\
&\quad < n_1 + \frac{n_1(n_1+1)}{2} + \frac{n_2(n_2+1)}{2} + n_1(n_2 - 1)\int_0^\infty \min\left\{1, \frac{\omega_2 + y_1}{\alpha_2}\right\} f_{21}(y)dy \\
&\overset{\text{Fact 1}}{=} n_1 + \frac{n_1(n_1+1)}{2} + \frac{n_2(n_2+1)}{2} + n_1(n_2 - 1)\left[\frac{\omega_2}{\alpha_2 - 1} - \frac{1}{\alpha_2 - 1}\left(\frac{\omega_2}{\alpha_2}\right)^{\alpha_2}\right].
\end{aligned} \tag{6}$$

Recall that by assumption $\omega_2 = \alpha_2 - 1$. Combining (6) and Fact 2, and letting α_2 tend to 1 from above, we find

$$\lim_{\alpha_2 \downarrow 1} \mathbf{E}\left[\Pi^*(n_1, n_2, \boldsymbol{\omega}, \boldsymbol{\alpha})\right] < n_1 + \frac{n_1(n_1+1)}{2} + \frac{(n_2+1)n_2}{2}.$$

Hence, it follows that for any n_1, n_2, $\omega_1 < \alpha_1 - 1$, there exists for any $\epsilon > 0$ an $\alpha^* > 1$ such that for all $1 < \alpha_2 = \omega_2 + 1 < \alpha^*$

$$\mathbf{E}\left[\Pi^*(n_1, n_2, \boldsymbol{\omega}, \boldsymbol{\alpha})\right] < n_1 + \frac{n_1(n_1+1)}{2} + \frac{n_2(n_2+1)}{2} + \epsilon.$$

As a straightforward consequence of Lemmata 3 and 4, we obtain the following theorem.

Theorem 3. *For any n_1 and n_2, there exist parameter settings $\boldsymbol{\omega} > \mathbf{0}$ and $\boldsymbol{\alpha} > \mathbf{1}$, such that, for any $\epsilon > 0$*

$$\frac{\mathbf{E}\left[\Pi^s(n_1, n_2, \boldsymbol{\omega}, \boldsymbol{\alpha})\right]}{\mathbf{E}\left[\Pi^*(n_1, n_2, \boldsymbol{\omega}, \boldsymbol{\alpha})\right]} > \frac{n_1^2 + n_2^2 + 2n_1n_2 + n_1 + n_2}{n_1^2 + n_2^2 + 3n_1 + n_2} - \epsilon.$$

Furthermore, there exist parameter settings $n_1, n_2 \geq 0$, $\boldsymbol{\omega} > \mathbf{0}$ and $\boldsymbol{\alpha} > \mathbf{1}$, such that for any $\epsilon > 0$,

$$\frac{\mathbf{E}\left[\Pi^s(n_1, n_2, \boldsymbol{\omega}, \boldsymbol{\alpha})\right]}{\mathbf{E}\left[\Pi^*(n_1, n_2, \boldsymbol{\omega}, \boldsymbol{\alpha})\right]} > 2 - \epsilon.$$

Proof. The restrictions imposed on the values α_1 and α_2 in Lemmas 3 and 4 can be satisfied simultaneously. Therefore, the first part of the theorem follows directly from these lemmas. To see the second part, we set $n_1 = n_2 = n$ and let n tend to infinity.

$$\lim_{n \to \infty} \frac{\mathbf{E}\left[\Pi^s(n, n, \boldsymbol{\omega}, \boldsymbol{\alpha})\right]}{\mathbf{E}\left[\Pi^*(n, n, \boldsymbol{\omega}, \boldsymbol{\alpha})\right]} > \lim_{n \to \infty} \frac{2n^2 + n}{n^2 + 2n} - \epsilon = \lim_{n \to \infty} \frac{2n+1}{n+2} - \epsilon = 2 - \epsilon.$$

4.2 Lower Bound on the Performance Guarantee of ℓ-SEPT

Similarly to the previous section, we show that for any $\epsilon > 0$ there exists an instance for which the ratio of the value of ℓ-SEPT to the value of OPT is only an additive ϵ away from the performance guarantee of Theorem 2.

Theorem 4. *There exist parameter settings $n_1, n_2 \geq 0$, $\boldsymbol{\omega} > \mathbf{0}$, $\boldsymbol{\alpha} > \mathbf{1}$ such that*

$$\frac{\mathbf{E}\left[\Pi^\ell(n_1, n_2, \boldsymbol{\omega}, \boldsymbol{\alpha})\right]}{\mathbf{E}\left[\Pi^*(n_1, n_2, \boldsymbol{\omega}, \boldsymbol{\alpha})\right]} > \frac{n_1^2 + n_2^2 + 2n_1n_2 + n_1 + n_2}{n_1^2 + n_2^2 + 3n_1 + n_2} - \epsilon$$

for any $\epsilon > 0$.

A formal proof of this theorem is given in the full version of the paper. In order to give this proof, we need a lower bound on the performance of ℓ-SEPT. To obtain this bound, we adjust the worst case instance of SEPT, given in Lemma 3. In that instance, we set our parameters in such a way that $\mathbf{E}[X_1]$ is slightly less than $\mathbf{E}[X_2]$. Hence, SEPT starts processing all jobs of class J_1, followed by the jobs of class J_2. OPT however, starts processing a job from class J_2, since the distribution of Θ_2 is flat, making it is beneficial to process a few jobs of the second class to get a better idea about the value of ϑ_2.

To create a bad instance for ℓ-SEPT, we would like to keep the same structure. Therefore, we need to make sure ℓ-SEPT does not switch to processing jobs from the second class after it processed a few jobs of the first class. This is done by setting the values of ω_1 and α_1 extremely large such that we are almost certain about the value of ϑ_1. Consequently, the realizations of processing times of jobs from class J_1 barely affect the expected processing time for the next job to be processed, i. e., when ω_1 and α_1 are big enough we have

$$\frac{\omega_1 + \sum_{j=1}^{k} x_k}{\alpha_1 + k - 1} \approx \frac{\omega_1}{\alpha_1 - 1} = 1 - \epsilon < 1 = \frac{\omega_2}{\alpha_2 - 1}$$

after k observations on the first job class.

5 Computational Results

In this section, we present preliminary computational results to investigate the performance of SEPT and ℓ-SEPT with respect to the optimal value in a Bayesian setting. That is, for several job class settings, we compare the values of SEPT and ℓ-SEPT with the optimal Bayesian solution. All computations are performed in MATLAB. In order to compute the values of OPT, we used the algorithm presented in Section 4 of the paper of Hamada and Glazebrook [11].

The Bayesian scheduling instances studied are as follows: the number of jobs in both job classes is set to 15, since the theoretical worst-case performance is reached for equal number of jobs in both classes. Furthermore, the gamma prior settings are set such that ω_i and $(\alpha_i - 1)$ are both an element of $\{0.5\,;\,1.0\,;\,5.0\,;\,25.0\}$ for each job class J_i. This results in 100 different computations covering the majority of interesting job class settings, i.e., the cases in which both job classes have high or low parameter uncertainty, and the mixed case in which one class has high and the other one low parameter uncertainty. Moreover, these computations could still be performed in a reasonable amount of time. Choosing our settings in a more extreme fashion immediately results in difficulties with the precision in calculating OPT, and also significantly increases the computation time of this optimal policy.

In our computations, 50.000 simulations are run for each Bayesian scheduling instance. In each of those simulations, we draw for each job class a parameter realization from a gamma distribution. This realization is subsequently used to draw 15 processing time realizations from an exponential distribution. Using these realizations the sum of completion times for each of the policies is calculated. Performance of the policies SEPT and ℓ-SEPT is measured by average objective value of the policy over the average objective value of OPT.

The preliminary computational results indicate that in case both job classes have high parameter uncertainty ℓ-SEPT is only about 1% away from the optimal value, while for SEPT the deviation is more than 13%. On the other hand, when the parameter uncertainty is low, we find that SEPT performs already better (1% away from OPT), but ℓ-SEPT obtains exactly the same value as OPT. In case both job classes have the same expected processing time, SEPT has the worst performance ratio among the instances tested: for high parameter uncertainty SEPT is about 30% above OPT, and for medium parameter uncertainty it is still 7% away from the optimal value. Intuitively, this was to be expected, because in these cases SEPT will just randomly choose a job class to start with. Also ℓ-SEPT performs the worst when both job classes have the same expected processing time, and in addition one job class has high parameter uncertainty, whereas the other one has low parameter uncertainty. This is explained by the fact that ℓ-SEPT makes its decisions based only on the first moment of the distribution and disregards further moments. Still in these cases, ℓ-SEPT outperforms SEPT, and it has a maximum deviation from OPT of only 9%. To conclude, on all instances ℓ-SEPT clearly outperforms the non-adaptive variant SEPT, thereby emphasizing the impact of learning on the performance of the algorithm. Finally, we remark that when averaging over the 50.000 trials, SEPT has a much higher variance than the other two policies. Again this is explained by the fact that SEPT, in case that the two job classes have the same expected processing time, randomly picks a job class to start with.

6 Concluding Remarks

In this paper, we studied the performance guarantee of two natural extensions of the traditional stochastic scheduling policy SEPT to the setting of Bayesian scheduling. We only considered the case in which there are 2 job classes and gave tight performance guarantees for both policies. An interesting extension will be the case of m job classes. For this case, we can prove a performance guarantee of m on both SEPT and ℓ-SEPT. For the non-adaptive policy SEPT this bound is tight, whereas for the adaptive policy ℓ-SEPT, we have a lower bound of $1 + \sqrt{m-1}$ and we conjecture that this is the right performance guarantee.

Acknowledgments. We thank three anonymous reviewers for their helpful comments to improve the exposition of the paper.

References

1. Araman, V.F., Caldentey, R.: Dynamic pricing for nonperishable products with demand learning. Operations Research 57(5), 1169–1188 (2009)
2. Burnetas, A.N., Katehakis, M.N.: On sequencing two types of tasks on a single processor under incomplete information. Probability in the Engineering and Informational Sciences 7(1), 85–119 (1993)

3. Chen, L., Plambeck, E.L.: Dynamic inventory management with learning about the demand distribution and substitution probability. Manufacturing & Service Operations Management 10(2), 236–256 (2008)
4. Dean, B.C.: Approximation Algorithms for Stochastic Scheduling Problems. PhD thesis, Massachusetts Institute of Technology (2005)
5. DeGroot, M.H.: Optimal Statistical Decisions. McGraw-Hill, New York (1970)
6. Farias, F.F., Van Roy, B.: Dynamic pricing with a prior on market response. Operations Research 58(1), 16–29 (2010)
7. Gittins, J.C.: Multi-armed bandit allocation indices. Wiley, N.Y. (1989)
8. Gittins, J.C., Glazebrook, K.D.: On Bayesian models in stochastic scheduling. Journal of Applied Probability 14(3), 556–565 (1977)
9. Gittins, J.C., Jones, D.M.: A dynamic allocation index for the sequential design of experiments. In: Progress in Statistics, pp. 241–266 (1974)
10. Glazebrook, K.D., Owen, R.W.: On the value of adaptive solutions to stochastic scheduling problems. Mathematics of Operations Research 20(1), 65–89 (1995)
11. Hamada, T., Glazebrook, K.D.: A Bayesian sequential single machine scheduling problem to minimize the expected weighted sum of flowtimes of jobs with exponential processing times. Operations Research 41(5), 924–934 (1993)
12. Hamada, T., Tamaki, M.: Some results on a Bayesian sequential scheduling on two identical parallel processors. Journal of the Operations Research Society of Japan 42(14), 316–329 (1999)
13. Lariviere, M.A., Porteus, E.L.: Stalking information: Bayesian inventory management with unobserved lost sales. Management Science 45(3), 346–363 (1999)
14. Lin, K.Y.: Dynamic pricing with real-time demand learning. Operations Research 174(1), 522–538 (2003)
15. Megow, N., Uetz, M., Vredeveld, T.: Models and algorithms for stochastic online scheduling. Mathematics of Operations Research 31(3), 513–525 (2006)
16. Megow, N., Vredeveld, T.: Approximation in Preemptive Stochastic Online Scheduling. In: Azar, Y., Erlebach, T. (eds.) ESA 2006. LNCS, vol. 4168, pp. 516–527. Springer, Heidelberg (2006)
17. Möhring, R.H., Radermacher, F.J., Weiss, G.: Stochastic scheduling problems I: General strategies. ZOR – Zeitschrift für Operations Research 28, 193–260 (1984)
18. Möhring, R.H., Schulz, A.S., Uetz, M.: Approximation in stochastic scheduling:the power of LP-based priority policies. Journal of ACM 46(6), 924–942 (1999)
19. Rothkopf, M.H.: Scheduling with random service times. Management Science 12(9), 703–713 (1966)
20. Scarf, H.: Bayes solutions of the statistical inventory problem. The Annals of Mathematical Statistics 30(2), 490–508 (1959)
21. Schulz, A.S.: New old algorithms for stochastic scheduling. In: Algorithms for Optimization with Incomplete Information (2005); Dagstuhl Seminar Proceedings, vol. 05031
22. Smith, W.E.: Various optimizers for single stage production. Naval Research Logistics Quaterly 3, 59–66 (1956)
23. Weiss, G.: Approximation results in parallel machines stochastic scheduling. Annals of Operations Research 26(1), 195–242 (1990)
24. Weiss, G.: Turnpike optimality of Smith's rule in parallel machines stochastic scheduling. Mathematics of Operations Research 17(2), 255–270 (1992)