Zuse Institute Berlin

Takustr. 7
14195 Berlin
Germany

GUILLAUME SAGNOL, DANIEL SCHMIDT GENANNT WALDSCHMIDT,
ALEXANDER TESCH

The Price of Fixed Assignments in Stochastic Extensible Bin Packing

This research is carried out in the framework of MATHEON supported by Einstein Foundation Berlin.

ZIB Report 18-19 (April 2018)

Zuse Institute Berlin
Takustr. 7
14195 Berlin
Germany

Telephone: +49 30-84185-0
Telefax: +49 30-84185-125

E-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

The Price of Fixed Assignments in Stochastic Extensible Bin Packing

Guillaume Sagnol[1,2], Daniel Schmidt genannt Waldschmidt[1], Alexander Tesch[2]

April 23, 2018

Abstract

We consider the *stochastic extensible bin packing problem* (SEBP) in which n items of stochastic size are packed into m bins of unit capacity. In contrast to the classical bin packing problem, bins can be extended at extra cost. This problem plays an important role in stochastic environments such as in surgery scheduling: Patients must be assigned to operating rooms beforehand, such that the regular capacity is fully utilized while the amount of overtime is as small as possible.

This paper focuses on essential ratios between different classes of policies: First, we consider the price of non-splittability, in which we compare the optimal non-anticipatory policy against the optimal fractional assignment policy. We show that this ratio has a tight upper bound of 2. Moreover, we develop an analysis of a fixed assignment variant of the LEPT rule yielding a tight approximation ratio of $(1 + e^{-1}) \approx 1.368$ under a reasonable assumption on the distributions of job durations. Furthermore, we prove that the price of fixed assignments, which describes the loss when restricting to fixed assignment policies, is within the same factor. This shows that in some sense, LEPT is the best fixed assignment policy we can hope for.

1 Stochastic Extensible Bin Packing

In the *extensible bin packing problem* (EBP), we must put n items of size $(p_1, \dots, p_n)$ in m bins, where the bins can be extended to hold more than the regular unit capacity. The cost of a bin is its final size: Specifically, a bin holding the items $I \subseteq \{1, \dots, n\}$ has a cost of $\max\big(\sum_{i \in I} p_i, 1\big)$. The goal is to minimize the total cost of the m bins.

The model of extensible bin packing naturally arises in scheduling problems with machines available for some amount of time at a fixed cost, and an additional cost for extra-time. So we stick to the scheduling terminology in this article (bins are machines, items are jobs, and item sizes are processing times). Recently, the model of EBP was adopted to handle surgery scheduling problems [8, 17, 2]: here, the machines are operating rooms, and the jobs are operations to be performed on patients. The extension of the regular working time of a machine corresponds to overtime for the medical staff. This application to surgery scheduling motivates the present paper: in practice, the duration of a surgical operation on a given patient is not known with certainty. Therefore, we want to study the stochastic counterpart of the extensible bin packing problem, in

[1]Technische Universität Berlin, Fakultät II, Institut für Mathematik, MA 5-2, Straße des 17. Juni 136, 10623 Berlin, Germany

[2]Zuse Institute Berlin Takustr. 7, 14195 Berlin, Germany

which the processing durations p_j's are only known probabilistically, and the expected cost of the machines is to be minimized.

Related work. EBP is closely related to another scheduling problem, where each job has a deadline d_j and the goal is to minimize the *total tardiness*. This problem can not be approximated within any constant factor in polynomial time, unless $P = NP$ [12]. Therefore, several articles studied approximation algorithms for a modified tardiness criterion, $\sum T_j + d_j$; see [11, 13]. The situation is very similar for extensible bin packing: the problem of minimizing the amount by which bins have to be extended is not approximable, and the criterion of EBP is obtained by adding the constant m to the objective.

The (deterministic version of) EBP was introduced by [6], who showed that the problem is strongly NP-hard, by reducing from 3–PARTITION; cf. [10]. Moreover, they prove that the *longest processing time first* (LPT) algorithm –which considers the jobs sorted in nonincreasing order of their processing time and assigns them sequentially to the machine with the largest remaining capacity– is a $\frac{13}{12}-$approximation algorithm. For equal bins, LPT can also be interpreted as iteratively assigning the jobs to the machine with the currently smallest load. In [7] the LPT algorithm was shown to be a $2(2-\sqrt{2}) \simeq 1.1716-$approximation algorithm for the case of unequal bin sizes. In a more general framework, Alon et. al. present a polynomial time approximation scheme [1].

The online version of the problem also attracted attention. Here, the jobs arrive one at a time and they must be assigned to a machine irrevocably. The list scheduling algorithm LS that assigns an incoming job to the machine with the largest remaining capacity was shown to have a competitive ratio of $\frac{5}{4}$ for equal bin sizes in [7] and was generalized in [21] for the case with unequal bin sizes. Furthermore, it was proven that no algorithm can achieve a performance of $\frac{7}{6}$ or smaller compared to the offline optimum. An improved online algorithm with a competitive ratio of 1.228 was also presented in [21].

In the context of surgery scheduling, a slightly more general framework has been introduced in [8]: the decision maker also chooses the number of bins of size S to open, at a fixed cost c^f, and there is a variable cost c^v for each minute of overtime. It is observed in [2] that every $(1+\rho)-$approximation algorithm for EBP yields a $(1+\rho\frac{Sc^v}{c^f})$-approximation algorithm in this more general setting. They also consider a two-stage stochastic variant of the problem, in which emergency patients should be allocated to operating rooms with pre-allocated elective patients. For this problem (in the case $S = c^v = c^f = 1$), a particular fixed assignment policy was shown to be a $\frac{5\theta}{4}$-approximation algorithm, when each job has a duration with bounded support $P_j \in [0, p_j^{\max}]$ such that $p_j^{\max} \leq \theta\mathbb{E}[P_j]$. To the best of our knowledge, this has been the only attempt to consider stochastic jobs in the literature on EBP.

In the remaining of this section, we introduce the *stochastic extensible bin packing problem* (SEBP). Throughout, we consider the problem of scheduling n stochastic jobs on m parallel identical machines non-preemptively. The set of machines and jobs are denoted by $\mathcal{M} = \{1, \ldots, m\}$ and $\mathcal{J} = \{1, \ldots, n\}$, respectively.

Stochastic Scheduling. Now, we want to give the intuition and main ideas of the required background in the field of stochastic scheduling. Precise definitions are given in [16]. The processing times are represented by a vector $P = (P_1, \ldots, P_n)$ of random variables. We denote by $\boldsymbol{p} = (p_1, \ldots, p_n) \in \mathbb{R}^n_{\geq 0}$ a particular realization of P. We assume that the P_j's are mutually independent, and that each processing time has a finite expected value. Unlike the deterministic case, a scheduling strategy can take more general forms than just an allocation of jobs to machines, as information is gained during the execution of the schedule. Indeed, job durations become known upon completion, and adaptive policies can react to the processing times observed so far.

We define a *schedule* as a pair $S = (\boldsymbol{s}, \boldsymbol{a}) \in \mathbb{R}^n_{\geq 0} \times \mathcal{M}^n$, where $s_j \geq 0$ is the starting time of job j and $a_j \in \mathcal{M}$ is the machine to which job j is assigned. A schedule S is said to be *feasible* for the realization $\boldsymbol{p}$ if each machine processes at most one job at a time:

$$\forall i \in \mathcal{M},\ \forall t \geq 0, \quad \Big|\{j \in \mathcal{J} :\ a_j = i,\ s_j \leq t < s_j + p_j\}\Big| \leq 1.$$

We denote by $\mathcal{S}(\boldsymbol{p})$ the set of all feasible schedules for the realization $\boldsymbol{p}$. A *planning rule* is a function Π that maps a vector $\boldsymbol{p} \in \mathbb{R}^n_{\geq 0}$ of processing times to a schedule $S \in \mathcal{S}(\boldsymbol{p})$. A planning rule is called a *scheduling policy* if it is *non-anticipatory*, which intuitively means that decisions taken at time t (if any) may only depend on the observed durations of jobs completed before t, and the probability distribution of the other processing times (conditioned by the knowledge that ongoing jobs have not completed before t).

Stochastic Extensible Bin Packing (SEBP). For a scheduling policy Π, we denote by S^Π_j and A^Π_j the random variables for the starting time of job j, and the machine to which j is assigned, respectively. The completion time of job j is $C^\Pi_j = S^\Pi_j + P_j$. We further introduce the random variable W^Π_i for the completion time of machine i, which is defined as the latest completion time of a job on machine i:

$$W^\Pi_i := \max\{C^\Pi_j |\ j \in \mathcal{J},\ A^\Pi_j = i\}.$$

It is easy to see that when Π is *non-idling*, i.e., if the starting time of any job is either 0 or equal to the completion time of the previous job assigned to the same machine, then

$$W^\Pi_i = \sum_{\{j \in \mathcal{J}:\ A^\Pi_j = i\}} P_j.$$

The realizations of the random vectors S^Π, A^Π, C^Π and W^Π for a vector of processing times $\boldsymbol{p}$ are denoted by appending $\boldsymbol{p}$ as an argument. For example, the workload of machine i for a non-idling policy Π in the scenario $\boldsymbol{p} \in \mathbb{R}^n_{\geq 0}$ is

$$W^\Pi_i(\boldsymbol{p}) = \sum_{j \xrightarrow{\Pi(\boldsymbol{p})} i} p_j,$$

where $j \xrightarrow{\Pi(\boldsymbol{p})} i$ means that $\Pi(\boldsymbol{p})$ assigns job j to machine i, i.e., we sum over indices $\{j \in \mathcal{J} : A^\Pi_j(\boldsymbol{p}) = i\}$.

Remark 1.1. We want to point out that other authors (e.g., in [1]) use the notation C_i for the machine completion times. We prefer to use the symbol W_i (which stands for *workload* in the non-idling case) to avoid the risk of confusion with the job completion times C_j.

We assume that jobs are scheduled on machines with an extendable working time, each machine having a unit regular working time. The cost incurred on machine i is equal to $\max(W^\Pi_i, 1)$, which accounts for the fixed costs, plus the amount by which the regular working time has to be extended. We are interested in strategies that minimize the expected value of the sum of costs over all machines:

$$\Phi(\Pi) := \mathbb{E}\Big[\ \sum_{i \in \mathcal{M}} \max(W^\Pi_i, 1)\ \Big].$$

The criterion can also be defined realization-wise: we define $\phi\big(\Pi, \boldsymbol{p}\big) := \sum_{i \in \mathcal{M}} \max(W^\Pi_i(\boldsymbol{p}), 1)$, so that $\Phi(\Pi) := \mathbb{E}_P[\phi\big(\Pi, P\big)]$.

Classes of scheduling policies. We define the following classes of scheduling policies:

- $\mathcal{P}$ denotes the class of all scheduling policies (non-anticipatory planning rules).
- $\mathcal{F}$ denotes the set of all *non-idling fixed-assignment policies*. Such policies are characterized by a vector of job-to-machine assignments $\boldsymbol{a} \in \mathcal{M}^n$, so that $A^\Pi(\boldsymbol{p}) = \boldsymbol{a}$ does not depend on the realization of processing times. For such a policy Π, it holds

$$\Phi(\Pi) = \sum_{i \in \mathcal{M}} \mathbb{E}\Big[\max\big(\sum_{j \xrightarrow{\Pi} i} P_j, 1\big)\Big],$$

  where the sum indexed by " $j \xrightarrow{\Pi} i$ " goes over all jobs j such that $A_j^\Pi = i$.

In addition, we define the following class of fractional policies (which cannot be considered as non-anticipatory planning rules, but will be useful to derive bounds):

- $\mathcal{R}$ denotes the class of fractional assignment policies, in which a fraction $a_{ij} \in [0,1]$ of job j is to be executed on machine i, with $\sum_{i \in \mathcal{M}} a_{ij} = 1$, for all $j \in \mathcal{J}$. For a "policy" $\Pi \in \mathcal{R}$, the different fractions of a job can be executed simultaneously on different machines, so

$$\Phi(\Pi) := \sum_{i \in \mathcal{M}} \mathbb{E}\Big[\max\big(\sum_{j \in \mathcal{J}} a_{ij}^\Pi P_j, 1\big)\Big].$$

LEPT policies. There is no unique way to generalize the LPT algorithm used in the deterministic case. We distinguish two variants of the "longest expected processing time first" (LEPT) policy. The policy $\text{LEPT}_\mathcal{F}$ is the fixed assignment policy that results in the same assignments as the LPT algorithm for the deterministic processing times $p_j = \mathbb{E}[P_j]$. In other words, job to machine assignments are precomputed offline, as follows: jobs are considered in decreasing order of $\mathbb{E}[P_j]$, and sequentially assigned to the least loaded machine (in expectation). An example of $\text{LEPT}_\mathcal{F}$ is depicted in Figure 1. The second policy, which we denote by $\text{LEPT}_\mathcal{P}$, is the priority list policy which considers jobs in the order of decreasing $\mathbb{E}[P_j]$'s, and start them (in this order) as early as possible. Unlike $\text{LEPT}_\mathcal{F}$, the job to machine assignments of the list policy $\text{LEPT}_\mathcal{P}$ depend on the realization $\boldsymbol{p}$ of the processing times. By [21] it immediately follows that $\text{LEPT}_\mathcal{P}$ is a $\frac{5}{4}$-approximation with respect to $OPT_\mathcal{P}$, since in every realization the schedule produced by $\text{LEPT}_\mathcal{P}$ is obtained by list scheduling.

In the remaining of this article, we focus on the policy $\text{LEPT}_\mathcal{F}$, which is more relevant in the context of surgery scheduling [8, 17, 2]. Indeed, fixed assignments yield more stable schedules and are better suited to handle the human resources of an operating theatre [9].

Performance ratios. For a given instance $I = (P, m)$ of the SEBP, we denote the optimum value in the class $\mathcal{C}$ of scheduling policies by

$$OPT_\mathcal{C}(I) = \inf_{\Pi \in \mathcal{C}} \Phi(\Pi).$$

Whenever the instance is clear from the context, or when $I = (P, m)$ is an arbitrary instance, we will drop I from the argument, so we simply write $OPT_\mathcal{C}$. We also denote by $OPT(\boldsymbol{p})$ the optimal value of the criterion for the deterministic problem with processing times $\boldsymbol{p}$. In this case, it is clear that we can restrict our attention to fixed assignment policies $\Pi \in \mathcal{F}$:

$$OPT(\boldsymbol{p}) = \inf_{\Pi \in \mathcal{F}} \phi(\Pi, \boldsymbol{p}).$$

We now define various performance ratios. We say that $\Pi \in \mathcal{C}$ is an *α-approximation in the class $\mathcal{C}$* if the inequality $\Phi(\Pi) \leq \alpha\, OPT_\mathcal{C}$ holds for all instances of SEBP. The *price of fixed assignments*

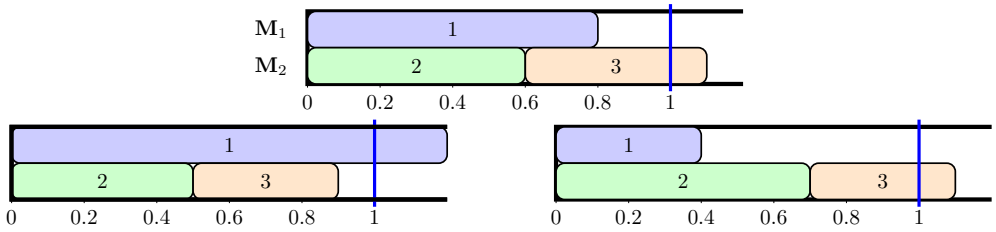

Figure 1: Example of a fixed assignment policy: assume machines $\mathcal{M} = \{1, 2\}$ and jobs $\mathcal{J} = \{1, 2, 3\}$ with processing time distributions $p_1 \in \{0.4, 1.2\}$, $p_2 \in \{0.5, 0.7\}$, $p_3 \in \{0.4, 0.6\}$ where each duration is attained with probability $\frac{1}{2}$. Since $\mathbb{E}[P_1] = 0.8 \geq \mathbb{E}[P_2] = 0.6 \geq \mathbb{E}[P_3] = 0.5$, $\text{LEPT}_{\mathcal{F}}$ assigns the jobs in order $1 \to 2 \to 3$ to the machines before their realization is known. The figure on the top depicts the resulting job to machine assignments with the average durations. For the realization $\boldsymbol{p} = (1.2, 0.5, 0.4)$ (lower left), $\text{LEPT}_{\mathcal{F}}$ is optimal with cost 2.2. For the realization $\boldsymbol{p} = (0.4, 0.7, 0.4)$ (lower right), $\text{LEPT}_{\mathcal{F}}$ yields cost 2.1. In contrast, note that $\text{LEPT}_{\mathcal{P}}$ would have started job 3 on the first machine after completion of job 1, giving a cost of 2.

and the *price of non-splittability* are respectively defined by

$$\text{PoFA} = \sup_I \frac{OPT_{\mathcal{F}}(I)}{OPT_{\mathcal{P}}(I)} \qquad \text{and} \qquad \text{PoNS} = \sup_I \frac{OPT_{\mathcal{P}}(I)}{OPT_{\mathcal{R}}(I)},$$

where the suprema go over all instances $I = (P, m)$ of SEBP. PoFA describes the loss if we restrict our attention to fixed assignment policies. In other words, it is a measure of what can be gained by allowing the use of more flexible, adaptive policies. This ratio has already gained some attention, e.g., in [15] and [19], whereby the latter shows that it can be arbitrarily large for the objective of minimizing the expected sum of completion times on parallel identical machines as the coefficient of variation grows. The second ratio (PoNS) is related to the power of preemption, see e.g. [4, 18, 5, 20], but should not be mixed up with it, because the class $\mathcal{R}$ allows different parts of a job to be processed simultaneously on several machines for fractional assignment policies.

Organization and Main results. Our paper is organized as follows. Section 2 deals with the price of non-splittability. We show that the expected cost of an optimal non-anticipatory policy is at most twice the expected cost of an optimal fractional assignment policy. Moreover, we present instances that achieve a lower bound arbitrarily close to 2, showing that PoNS = 2. In Section 3, we consider the case of short jobs ($P_j \in [0, 1]$ almost surely) and we obtain a performance guarantee of $1 + e^{-1}$ for $\text{LEPT}_{\mathcal{F}}$ compared to the stochastic optimum. This result is used in Section 4 to show that the price of fixed assignments is at most $1 + e^{-1}$, even without the restriction to instances with short jobs. We also give a family of instances where this bound is attained at the limit, which proves that $\text{PoFA} = 1 + e^{-1}$. In Section 5, we give preliminary results and conjecture that $\text{LEPT}_{\mathcal{F}}$ is a $1 + e^{-1}$-approximation algorithm, even when job durations may exceed 1. Finally, we show in Section 6 that the performance of $\text{LEPT}_{\mathcal{F}}$ can not be better than $\frac{4}{3}$ in the class $\mathcal{F}$.

2 The price of non-splittability

Proposition 2.1. *Let (P, m) be an instance of SEBP and let $\rho := \frac{1}{m}\sum_{j\in\mathcal{J}} \mathbb{E}[P_j]$ be the expected workload averaged over all machines. Then the following holds:*

$$OPT_{\mathcal{F}} \geq OPT_{\mathcal{P}} \geq \mathbb{E}_P[OPT(P)] \geq OPT_{\mathcal{R}} = \mathbb{E}\Big[\max(\sum_{j\in\mathcal{J}} P_j, m)\Big] \geq m\max(\rho, 1).$$

Proof. The first inequality follows immediately since $\mathcal{F} \subseteq \mathcal{P}$.

Next, for all policies $\Pi \in \mathcal{P}$ and all realizations $\boldsymbol{p}$ it holds $\phi(\Pi, \boldsymbol{p}) \geq OPT(\boldsymbol{p})$, by definition of an optimal policy for the deterministic processing times $\boldsymbol{p}$. Taking the expectation on both sides yields the second inequality.

Before we go on to the next inequality, we first show that $OPT_{\mathcal{R}} = \mathbb{E}\Big[\max(\sum_{j\in\mathcal{J}} P_j, m)\Big]$. To do so we show that for any realization $\boldsymbol{p}$ an optimal fractional assignment policy assigns all jobs uniformly to all machines. More precisely, we show that $a_{ij} = \frac{1}{m}$ for all $i \in \mathcal{M}$ and $j \in \mathcal{J}$ solves the following problem of finding the optimal fractional assignment:

$$\underset{0\leq a_{ij}\leq 1}{\textbf{minimize}} \quad \sum_{i\in\mathcal{M}} \max(\sum_{j\in\mathcal{J}} a_{ij}p_j, 1), \qquad \text{such that} \quad \sum_{i\in\mathcal{M}} a_{ij} = 1, \quad \forall j \in \mathcal{J}. \tag{1}$$

A trivial lower bound on the optimal value of Problem (1) is $\max(\sum_{j\in\mathcal{J}} p_j, m)$. This is true since for any feasible fractional assignment $(a_{ij})_{i\in\mathcal{M},j\in\mathcal{J}}$, $\sum_{i\in\mathcal{M}} \max(\sum_{j\in\mathcal{J}} a_{ij}p_j, 1) \geq \sum_{i\in\mathcal{M}}\sum_{j\in\mathcal{J}} a_{ij}p_j = \sum_{j\in\mathcal{J}} p_j$, and similarly, $\sum_{i\in\mathcal{M}} \max(\sum_{j\in\mathcal{J}} a_{ij}p_j, 1) \geq \sum_{i\in\mathcal{M}} 1 = m$. Choosing all fractions to be $\frac{1}{m}$ we obtain $\sum_{i\in\mathcal{M}} \max(\sum_{j\in\mathcal{J}} \frac{1}{m}p_j, 1) = m \cdot \max(\sum_{j\in\mathcal{J}} \frac{1}{m}p_j, 1) = \max(\sum_{j\in\mathcal{J}} p_j, m)$ which exactly matches the lower bound and hence, it must be optimal. Since this holds for any realization we can take the expected value resulting into the desired identity.

In order to show $\mathbb{E}_P[OPT(P)] \geq OPT_{\mathcal{R}}$, we observe that for any realization $\boldsymbol{p}$, Problem (1) is the continuous relaxation of the problem with binary variables for finding the optimal assignments for the deterministic problem with processing times $\boldsymbol{p}$. Hence, by again taking expectations this yields the inequality.

Finally, the last inequality is Jensen's inequality applied to the convex function $x \mapsto \max(x, m)$. □

In the next proposition, which we prove in the appendix, we show the intuitive fact that among non-idling policies, the worst case is to assign all jobs to the same machine.

Proposition 2.2. *Let $\Pi \in \mathcal{P}$ be non-idling and let Π_1 be the fixed assignment policy that schedules all jobs on machine 1. Then, $\Phi(\Pi) \leq \Phi(\Pi_1)$.*

We show that any non-idling policy is a 2-approximation in the class of non-anticipatory policies (and hence in the class of fixed-assignment policies).

Proposition 2.3. *Let Π be* any *non-idling policy. Then,*

$$\Phi(\Pi) \leq 2\, OPT_{\mathcal{R}}.$$

Proof. Let Π be a non-idling policy and Π_1 be the naive fixed assignment policy in which all jobs are scheduled on one machine without idle time. Proposition 2.2 yields that $\Phi(\Pi) \leq \Phi(\Pi_1)$, and we have

$$\Phi(\Pi_1) = \mathbb{E}[\max(\sum_{j\in\mathcal{J}} P_j, 1)] + (m-1) \leq \mathbb{E}[\max(\sum_{j\in\mathcal{J}} P_j, m)] + m - 1.$$

We know that $m \leq \mathbb{E}[\max(\sum_{j\in\mathcal{J}} P_j, m)] = OPT_{\mathcal{R}}$ from Proposition 2.1, so we have

$$\Phi(\Pi) \leq \Phi(\Pi_1) \leq 2\, OPT_{\mathcal{R}} - 1 \leq 2\, OPT_{\mathcal{R}}.$$

□

Consequently, we are only interested in finding $\alpha-$approximation algorithms for $\alpha < 2$, since a $2-$approximation algorithm performs no better (in the worst case) than the naive policy that puts all jobs on a single machine.

The last proposition also shows that the price of non-splittability is upper bounded by 2. In fact, this bound is tight:

Theorem 2.4. *The price of non-splittability of SEBP is* PoNS = 2.

The proof relies on a technical lemma which is proved in the appendix:

Lemma 2.5. *Let $Y \sim$ Poisson(λ) for some $\lambda \in \mathbb{N}$. Then,*

$$\frac{1}{\lambda}\mathbb{E}\Big[\max(Y,\lambda)\Big] = 1 + \frac{e^{-\lambda}\lambda^{\lambda}}{\lambda!}.$$

Proof of Theorem 2.4. It follows from Propositions 2.1 and 2.3 that $OPT_{\mathcal{P}} \leq OPT_{\mathcal{F}} \leq 2OPT_{\mathcal{R}}$.

Let $\lambda \in \mathbb{N}$ and consider the instance I with $n = m \geq \lambda$ independent and identically distributed jobs in which the processing time of each job j takes the value $\frac{m}{\lambda}$ with probability $\frac{\lambda}{m}$ and 0 otherwise. In other words, for all $j \in \mathcal{J}$ we have $P_j \sim \frac{m}{\lambda}\,\text{Bernoulli}(\frac{\lambda}{m})$. As $n = m$, an optimal non-idling policy clearly assigns each job to a different machine. This yields

$$OPT_{\mathcal{P}}(I) = m \cdot \mathbb{E}[\max(P_1, 1)] = m \cdot \Big((1 - \frac{\lambda}{m}) \cdot 1 + \frac{\lambda}{m} \cdot \frac{m}{\lambda}\Big) = 2m - \lambda.$$

For the objective value of an optimal fractional assignment policy we can use Proposition 2.1. We will also use the fact that the sum of i.i.d. Bernoulli random variables is binomially distributed, i.e., $X := \frac{\lambda}{m} \cdot \sum_{j\in\mathcal{J}} P_j \sim \text{Binomial}(m, \frac{\lambda}{m})$. Moreover, it is folklore that X converges in distribution to $Y \sim \text{Poisson}(\lambda)$ as $m \to \infty$.

Therefore, we have $\frac{\lambda}{m}OPT_{\mathcal{R}}(I) = \frac{\lambda}{m}\mathbb{E}\Big[\max\Big(\sum_{j\in\mathcal{J}} P_j, m\Big)\Big] = \mathbb{E}[\max(X,\lambda)]$, which converges in distribution to $1 + \frac{e^{-\lambda}\lambda^{\lambda}}{\lambda!}$ as $m \to \infty$ by Lemma 2.5. Putting all together, the ratio $OPT_{\mathcal{P}}(I)/OPT_{\mathcal{R}}(I)$ converges to $2(1 + \frac{e^{-\lambda}\lambda^{\lambda}}{\lambda!})^{-1}$ as $m \to \infty$, and this quantity can be made arbitrarily close to 2 by choosing λ large enough. □

3 Approximation ratio of LEPT: The case of short jobs

In this section, we show that $LEPT_{\mathcal{F}}$ is an $(1 + e^{-1})$-approximation algorithm when the instance only contains *short jobs*.

Definition 3.1. We say job j is *short* if its processing time P_j is less than or equal to 1 almost surely, i.e.,

$$\mathbb{P}[0 \leq P_j \leq 1] = 1.$$

It is reasonable to assume that jobs are short: In real world applications, such as in surgery scheduling, the duration of a single operation rarely exceeds the regular capacity of an operating room. Moreover, this assumption is not uncommon; cf. [7, 21]. The proof of the performance guarantee of $LEPT_{\mathcal{F}}$ relies on three lemmas which we prove in the appendix. The first lemma gives a tight bound on the expected cost incurred on one machine.

Lemma 3.2. *Let k be some positive integer and let all jobs $j \in [k]$ be short. Then,*

$$\mathbb{E}\Big[\max\Big(\sum_{j=1}^{k} P_j, 1\Big)\Big] \leq \sum_{j=1}^{k} \mathbb{E}[P_j] \;+\; \prod_{j=1}^{k}(1 - \mathbb{E}[P_j]).$$

Moreover, this bound is tight, and attained for the two point distributions $P_j^* \sim \text{Bernoulli}(\mathbb{E}[P_j])$.

The second lemma gives bounds on the expected workload of any machine in an $\text{LEPT}_{\mathcal{F}}$ schedule. Interestingly, the gap between the lower and upper bounds becomes smaller when the number of jobs scheduled on a machine grows.

Lemma 3.3. *Let x_i denote the expected load of machine $i \in \mathcal{M}$ produced by LEPT$_\mathcal{F}$, i.e., $x_i := \mathbb{E}[W_i^{LEPT_\mathcal{F}}] = \sum_{j \xrightarrow{LEPT_\mathcal{F}} i} \mathbb{E}[P_j]$. Then, there exists $\ell \geq 0$ such that for all $i \in \mathcal{M}$,*

$$\ell \leq x_i \leq \frac{n_i}{n_i - 1}\ell,$$

where $n_i := \left|\{j \in \mathcal{J} : j \xrightarrow{LEPT_\mathcal{F}} i\}\right|$ denotes the number of jobs assigned to machine i, and we use the convention $\frac{n_i}{n_i - 1} = \frac{1}{0} := +\infty$ whenever $n_i = 1$.

We need a third lemma with a technical result:

Lemma 3.4. *Let $\ell \geq 0$ and $\rho \geq \ell$. We define the function $h : [0,1] \to \mathbb{R},\ y \mapsto (1-y)^{1+\frac{\ell}{y}}$, which is defined by continuity at $y = 0$ with $h(0) = e^{-\ell}$. Let $\boldsymbol{y} \in [0,1]^m$ be any vector satisfying the inequality $\sum_{i \in \mathcal{M}} y_i = m(\rho - \ell)$. Then,*

$$\sum_{i \in \mathcal{M}} h(y_i) \leq m e^{-\rho}.$$

We are now ready to prove the main result of this section:

Theorem 3.5. *Consider an instance (P, m) with only short jobs. Let $\rho := \frac{1}{m}\sum_{j \in \mathcal{J}} \mathbb{E}[P_j]$ denote the expected workload averaged over all machines. Then it holds*

$$\frac{\Phi(LEPT_\mathcal{F})}{m \max(\rho, 1)} \leq \frac{\rho + e^{-\rho}}{\max(\rho, 1)} \leq 1 + e^{-1}.$$

Proof. Let J_i denote the subset of jobs that LEPT$_\mathcal{F}$ assigns to machine $i \in \mathcal{M}$ and let $n_i := |J_i|$. As in Lemma 3.3, let LEPT$_\mathcal{F}$ produce an expected workload of $x_i = \sum_{j \in J_i} \mathbb{E}[P_j]$ on machine i. Then, by Lemma 3.2 we can bound the expected cost incurred on machine i as

$$\mathbb{E}[\max(W_i^{\mathrm{LEPT}_\mathcal{F}}, 1)] \leq \sum_{j \in J_i} \mathbb{E}[P_j] + \prod_{j \in J_i} (1 - \mathbb{E}[P_j]) \leq x_i + \left(1 - \frac{x_i}{n_i}\right)^{n_i} \tag{2}$$

where the last inequality follows from the Schur-concavity of $\boldsymbol{\mu} \mapsto \prod_{j \in J_i}(1 - \mu_j)$ over $[0,1]^{n_i}$; cf. [14, Proposition 3.E.1]. Next, we apply Lemma 3.3, so there exists an $\ell \geq 0$ such that $\ell \leq x_i \leq \frac{n_i}{n_i - 1}\ell$. Let $y_i := x_i - \ell \geq 0$. The second inequality can be rewritten as

$$n_i \leq \frac{x_i}{x_i - \ell} = 1 + \frac{\ell}{y_i}, \tag{3}$$

which remains valid for $y_i = 0$ if we define $\ell/0 := +\infty$. We know that $P_j \in [0,1]$ almost surely, in particular $\mathbb{E}[P_j] \leq 1$, and hence, $x_i \leq n_i$. For this reason, the above inequality implies $x_i \leq \frac{x_i}{x_i - \ell}$ and therefore, $y_i \leq 1$. By combining (2) and (3), and using the fact that $(1 - \frac{x_i}{n_i})^{n_i}$ is a nondecreasing function of n_i, we obtain

$$\mathbb{E}[\max(W_i^{\mathrm{LEPT}_\mathcal{F}}, 1)] \leq x_i + (1 - (x_i - \ell))^{\frac{x_i}{x_i - \ell}} = \ell + y_i + h(y_i), \tag{4}$$

where h is the function defined in Lemma 3.4, and the y_i's satisfy $y_i \in [0,1]$. Moreover, we have $\sum_{i \in \mathcal{M}} y_i = m(\rho - \ell) \iff \sum_{i \in \mathcal{M}}(\ell + y_i) = \sum_{i \in \mathcal{M}} x_i = \rho m$. Summing up the inequalities (4) over all $i \in \mathcal{M}$ and using Lemma 3.4 yields

$$\Phi(\mathrm{LEPT}_\mathcal{F}) = \sum_{i \in \mathcal{M}} \mathbb{E}[\max(W_i^{\mathrm{LEPT}_\mathcal{F}}, 1)] \leq \rho m + \sum_{i \in \mathcal{M}} h(y_i) \leq m(\rho + e^{-\rho}).$$

As a consequence, we obtain

$$\frac{\Phi(\mathrm{LEPT}_{\mathcal{F}})}{m\max(\rho,1)} \leq \frac{\rho + e^{-\rho}}{\max(\rho,1)}.$$

Finally, the second inequality of the theorem follows from the fact that the above ratio is maximized for $\rho = 1$. This is true because $\frac{\rho+e^{-\rho}}{\max(\rho,1)} = \rho + e^{-\rho}$ on $[0,1]$, hence increasing, and $\frac{\rho+e^{-\rho}}{\max(\rho,1)} = 1 + \frac{e^{-\rho}}{\rho}$ on $[1,+\infty]$, hence decreasing. □

Combining this result with the inequality $OPT_{\mathcal{P}} \geq m\max(1,\rho)$ from Proposition 2.1 yields the following

Corollary 3.6. *The $LEPT_{\mathcal{F}}$ policy is an $(1+e^{-1})$-approximation algorithm in the class $\mathcal{P}$, over the set of instances with short jobs only.*

As we will see in the next section, our analysis of $\mathrm{LEPT}_{\mathcal{F}}$ is tight. In section 5 we discuss how this result might be extended considering additionally long jobs.

4 The Price of Fixed Assignments

In this section, we are going to show that the price of fixed assignments is equal to $1 + e^{-1}$. To do this, we require a lemma that will allow us to focus on instances with short jobs. Our analysis relies on a parameter $\alpha \geq 0$ which quantifies the *length excess* of jobs (for an instance with only short jobs, it holds $\alpha = 0$).

Lemma 4.1. *Let $I = (P, m)$ be an instance of SEBP, and let $I' = (P', m)$ denote the instance in which the processing time P_j of all jobs is replaced by $P'_j = \min(P_j, 1)$. Let $\alpha = \sum_{j\in\mathcal{J}} \alpha_j$, where we define $\alpha_j := \mathbb{E}[\max(P_j - 1, 0)] \geq 0$. The new P'_js are short jobs, and we have*

$$OPT_{\mathcal{F}}(I') = OPT_{\mathcal{F}}(I) - \alpha \quad \text{and} \quad \mathbb{E}_{P'}[OPT(P')] = \mathbb{E}_P[OPT(P)] - \alpha.$$

Proof. Let J_i and J'_i denote the subsets of jobs assigned to machine i in an optimal fixed assignment policy Π for instance I, and in an optimal fixed assignment policy Π' for instance I', respectively. Let $\boldsymbol{p}$ be a realization of the processing times for instance I, and let $\boldsymbol{p}'$ denote the vector with elements $p'_j = \min(p_j, 1)$. We compute the difference between the costs incurred by $\Pi(\boldsymbol{p})$ and $\Pi(\boldsymbol{p}')$ on machine i:

$$\max(W_i^{\Pi}(\boldsymbol{p}), 1) - \max(W_i^{\Pi}(\boldsymbol{p}'), 1) = \max\Big(\sum_{j\in J_i} p_j, 1\Big) - \max\Big(\sum_{j\in J_i} \min(p_j, 1), 1\Big). \tag{5}$$

It is easy to see that $\sum_{j\in J_i} p_j \leq 1 \iff \sum_{j\in J_i} \min(p_j, 1) \leq 1$. Hence, we distinguish two cases. If $\sum_{j\in J_i} p_j \leq 1$, then the right hand side of (5) vanishes. Otherwise, the right hand side of (5) becomes $\sum_{j\in J_i} p_j - \min(p_j, 1) = \sum_{j\in J_i} \max(p_j - 1, 0)$. In both cases, it holds $\max(W_i^{\Pi}(\boldsymbol{p}), 1) - \max(W_i^{\Pi}(\boldsymbol{p}'), 1) = \sum_{j\in J_i} \max(p_j - 1, 0)$. Taking the expectation and summing up over all machines yields

$$\Phi_I(\Pi) - \Phi_{I'}(\Pi) = \sum_{i\in\mathcal{M}} \sum_{j\in J_i} \alpha_j = \sum_{j\in\mathcal{J}} \alpha_j = \alpha,$$

where the symbol $\Phi_I(\Pi)$ emphasizes that the expected value in the criterion is taken with respect to the processing time distributions of instance I.

Since Π is optimal in the class $\mathcal{F}$ for instance I, we have $\Phi_I(\Pi) = OPT_{\mathcal{F}}(I)$ and $\Phi_{I'}(\Pi) \geq OPT_{\mathcal{F}}(I')$. Hence,

$$\Phi_{I'}(\Pi) = OPT_{\mathcal{F}}(I) - \alpha \geq OPT_{\mathcal{F}}(I'). \tag{6}$$

Similarly, the comparison of the costs incurred by $\Pi'(\boldsymbol{p})$ and $\Pi'(\boldsymbol{p}')$ on machine i yields $\max(W_i^{\Pi'}(\boldsymbol{p}), 1) - \max(W_i^{\Pi'}(\boldsymbol{p}'), 1) = \sum_{j \in J_i'} \max(p_j - 1, 0)$. Again, by taking the expectation and summing over all machines we obtain $\Phi_I(\Pi') - \Phi_{I'}(\Pi') = \sum_{j \in \mathcal{J}} \alpha_j = \alpha$. Now, we observe that $\Phi_{I'}(\Pi') = OPT_{\mathcal{F}}(I')$ and $\Phi_I(\Pi') \geq OPT_{\mathcal{F}}(I)$, so we have

$$\Phi_I(\Pi') = OPT_{\mathcal{F}}(I') + \alpha \geq OPT_{\mathcal{F}}(I). \tag{7}$$

Finally, by combining (6) and (7) we obtain $OPT_{\mathcal{F}}(I) - \alpha \geq OPT_{\mathcal{F}}(I') \geq OPT_{\mathcal{F}}(I) - \alpha$, which shows the desired equality:

$$OPT_{\mathcal{F}}(I) - \alpha = OPT_{\mathcal{F}}(I').$$

The proof of the equality $\mathbb{E}_{P'}[OPT(P')] = \mathbb{E}_P[OPT(P)] - \alpha$ works in a similar manner, but we must take sums over a different subset of jobs $J_i(\boldsymbol{p})$ for each scenario $\boldsymbol{p}$, corresponding to the jobs that an optimal policy assigns to machine i for the deterministic problem with processing times $\boldsymbol{p}$. □

We can now prove the main result of this section:

Theorem 4.2. *The price of fixed assignments for SEBP is equal to* $(1 + e^{-1})$*:*

$$\text{PoFA} = 1 + e^{-1}.$$

Proof. Let $I = (P, m)$ denote an instance of SEBP and $I' = (P', m)$ the reduced instance as in Lemma 4.1. We have:

$$\frac{OPT_{\mathcal{F}}(I)}{OPT_{\mathcal{P}}(I)} \leq \frac{OPT_{\mathcal{F}}(I)}{\mathbb{E}_P[OPT(P)]} = \frac{OPT_{\mathcal{F}}(I') + \alpha}{\mathbb{E}_{P'}[OPT(P')] + \alpha} \leq \frac{OPT_{\mathcal{F}}(I')}{\mathbb{E}_{P'}[OPT(P')]} \leq 1 + e^{-1},$$

where the first inequality follows from Proposition 2.1, the equality is a consequence of Lemma 4.1, the second inequality follows from $\alpha \geq 0$, and the last inequality results from Proposition 2.1 and Theorem 3.5. Therefore, it remains to show that for all $\epsilon > 0$ there exists an instance I in which we have

$$\frac{OPT_{\mathcal{F}}(I)}{OPT_{\mathcal{P}}(I)} \geq 1 + e^{-1} - \epsilon.$$

For this purpose, we consider an instance $I = (P, m)$ in which we have $n = km$ jobs for some $k \in \mathbb{N}$, where $P_j \sim \text{Bernoulli}(\frac{1}{k})$ for all $j \in \mathcal{J}$. An optimal fixed assignment policy assigns each machine the same number of jobs, in this case k. The cost on one machine is hence the expected value of $\max(Z, 1)$, where $Z := \sum_{j=1}^{k} P_j \sim \text{Binomial}(k, \frac{1}{k})$. So,

$$\begin{aligned} OPT_{\mathcal{F}}(I) = m \cdot \mathbb{E}[\max(Z, 1)] &= m \cdot \Big(\mathbb{E}\big[Z | Z \geq 1\big] \ \mathbb{P}[Z \geq 1] + \mathbb{E}\big[1 | Z < 1\big] \ \mathbb{P}[Z < 1] \Big) \\ &= m \cdot \big(\mathbb{E}[Z] + \mathbb{P}[Z = 0] \big) = m \cdot \left(1 + \left(1 - \frac{1}{k} \right)^k \right), \end{aligned}$$

which converges to $m(1 + e^{-1})$ as $k \to \infty$. On the other hand, an optimal policy in $\mathcal{P}$ lets a job run whenever a machine becomes idle. The cost of an optimal policy is hence m whenever less than m jobs have duration 1, and is equal to $\sum_{j=1}^{km} p_j$ otherwise. This shows that $OPT_{\mathcal{P}}(I) =$

$\mathbb{E}[\max(U,m)]$, where $U := \sum_{j=1}^{km} P_j \sim \text{Binomial}\left(km, \frac{1}{k}\right)$. Now, we can argue as in Theorem 3.5 that U converges in distribution to $Y \sim \text{Poisson}(m)$ as $k \to \infty$. So, by Lemma 2.5, we have

$$OPT_{\mathcal{P}}(I) \to m \cdot \left(1 + \frac{e^{-m}m^m}{m!}\right) \quad \text{as} \quad k \to \infty.$$

Finally, we have shown that the ratio of $OPT_{\mathcal{F}}(I)$ to $OPT_{\mathcal{P}}(I)$ can be made arbitrarily close to $(1+e^{-1}) \cdot \left(1 + \frac{e^{-m}m^m}{m!}\right)^{-1}$ by choosing k large enough. We conclude by observing that $\lim_{m\to\infty} \frac{m^m e^{-m}}{m!} = 0$, so this ratio can be arbitrarily close to $1 + e^{-1}$. □

This proves that our analysis of $LEPT_{\mathcal{F}}$ is tight. It even shows that $LEPT_{\mathcal{F}}$ is the best fixed assignment policy in the following sense: Since there exists instances for which the ratio of an optimal fixed assignment policy to an optimal non-anticipatory policy is arbitrarily close to $1 + e^{-1}$ and the fact that $LEPT_{\mathcal{F}}$ is a $1 + e^{-1}$-approximation (for short jobs), we cannot hope to find a policy $\Pi \in \mathcal{F}$ with a better approximation guarantee in the class $\mathcal{P}$.

5 Extending the approximation guarantee of LEPT for Instances with long jobs

In this section, we discuss the possibility to extend the $(1 + e^{-1})$-approximation guarantee of $\text{LEPT}_{\mathcal{F}}$ for instances containing long jobs, that is, jobs whose duration may exceed 1. It can be shown –using a similar approach as in Theorem 3.5– that $\frac{\Phi(\text{LEPT}_{\mathcal{F}})}{OPT_{\mathcal{R}}} \leq 1 + e^{-\frac{1}{d_{\max}}}$ for instances where each job satisfies $P_j \in [0, d_{\max}]$ almost surely for some $d_{\max} \geq 1$, and that this bound is tight. Letting $d_{\max} \to \infty$ just gives the trivial approximation guarantee of 2, so we have to use a better lower bound on $OPT_{\mathcal{P}}$ in order to prove that $\text{LEPT}_{\mathcal{F}}$ is a $(1 + e^{-1})$-approximation algorithm.

Our next candidate is the bound $OPT_{\mathcal{P}} \geq \mathbb{E}_P[OPT(P)]$, cf. Proposition 2.1. We think that an analysis relying on the parameters $s = \sum_j \mathbb{E}[P_j]$ and $\alpha = \sum_j \mathbb{E}[\max(0, P_j - 1)]$ introduced in Lemma 4.1 could lead to the desired result. We did not manage to get a complete proof so far, but we present some preliminary results and we make a conjecture.

Throughout this section we use the following notation: P' represents the vector of processing times truncated above one, i.e. $P'_j = \min(P_j, 1)$. We denote by J_i the set of jobs assigned to machine i by $\text{LEPT}_{\mathcal{F}}$, and we let $n_i := |J_i|$. Without loss of generality we assume $n_i \geq 1$ $(\forall i \in \mathcal{M})$, as otherwise it is clear that $n < m$, and $\text{LEPT}_{\mathcal{F}}$ is an optimal policy.

Lemma 5.1. $E_P[OPT(P)] \geq \max(s, m + \alpha)$.

Proof. We already know from Lemma 4.1 that $E_P[OPT(P)] = E_{P'}[OPT(P')] + \alpha$. Then, the result follows from $E_{P'}[OPT(P')] \geq \max(\sum_{j\in\mathcal{J}} \mathbb{E}[P'_j], m)$, cf. Proposition 2.1, and the identity $\sum_{j\in\mathcal{J}} \mathbb{E}[P'_j] = \sum_{j\in\mathcal{J}} \mathbb{E}[P_j] - \alpha$, implying that $E_P[OPT(P)] \geq \max(s - \alpha, m) + \alpha = \max(s, m + \alpha)$. □

Lemma 5.2. $\Phi(LEPT_{\mathcal{F}}) = \sum_i \mathbb{E}[\max(\sum_{j\in J_i} P'_j, 1)] + \alpha$.

Proof. The cost on machine i for realization $\boldsymbol{p}$ is

$$\max(\sum_{j\in J_i} p_j, 1) = \max\Big(\sum_{j\in J_i} p'_j, 1\Big) + \sum_{j\in J_i} \max(0, p_j - 1),$$

where $p'_j := \min(p_j, 1)$, which can be seen by distinguishing between the cases where $\forall j \in J_i,\ p_j \leq 1$ or $\exists j \in J_i : p_j > 1$. The result follows from summing this equality over all machines and taking the expectation. □

Lemma 5.3. *Let $v^*(s,\alpha)$ denote the optimal value of the following optimization problem:*

$$\underset{\boldsymbol{x},\boldsymbol{\beta},\ell\geq 0}{\textbf{maximize}} \quad \sum_{i\in\mathcal{M}} \left(1-\frac{x_i}{n_i}\right)^{n_i} \tag{8a}$$

$$s.t. \quad \sum_{i\in\mathcal{M}} x_i = s-\alpha \tag{8b}$$

$$\sum_{i\in\mathcal{M}} \beta_i = \alpha \tag{8c}$$

$$\ell \leq x_i+\beta_i \leq \ell\frac{n_i}{n_i-1}, \qquad \forall i\in\mathcal{M}. \tag{8d}$$

Then, $\Phi(LEPT_{\mathcal{F}}) \leq s + v^*(s,\alpha)$.

Proof. Since the reduced jobs P'_j are short, we can insert the inequality of Lemma 3.2 into the result of Lemma 5.2, and we obtain:

$$\Phi(\mathrm{LEPT}_{\mathcal{F}}) \leq \sum_{j\in\mathcal{J}} \mathbb{E}[P'_j] + \sum_{i\in\mathcal{M}} \prod_{j\in J_i} \left(1-\mathbb{E}[P'_j]\right) + \alpha \leq s + \sum_{i\in\mathcal{M}} \left(1-\frac{\sum_{j\in J_i}\mathbb{E}[P'_j]}{n_i}\right)^{n_i},$$

where the second inequality follows from the identity $\sum_{j\in\mathcal{J}}\mathbb{E}[P'_j] = s-\alpha$ and the Schur-concavity of $\boldsymbol{\mu} \mapsto \prod_{j\in J_i}(1-\mu_j)$ (we already used this argument in the proof of Theorem 3.5). Then, we obtain the result by observing that $x_i = \sum_{j\in J_i}\mathbb{E}[P'_j]$, $\beta_i = \sum_{j\in J_i}\mathbb{E}[\max(P_j-1,0)]$, and $\ell = \min_{i\in\mathcal{M}}(x_i+\beta_i)$ is a feasible solution for the optimization problem in the lemma. Indeed, we have $\sum_{i\in\mathcal{M}} x_i = \sum_{j\in\mathcal{J}}\mathbb{E}[P'_j] = s-\alpha$, $\sum_{i\in\mathcal{I}}\beta_i = \sum_{j\in\mathcal{J}}\mathbb{E}[\max(P_j-1,0)] = \alpha$, and by Lemma (3.3), the last constraint holds, because $x_i+\beta_i$ is the expected load of the ith machine in an $\mathrm{LEPT}_{\mathcal{F}}$ schedule. □

As a consequence of the above lemmas, we observe that if the ratio $\frac{s+v^*(s,\alpha)}{\max(s,m+\alpha)}$ is bounded from above by $1+e^{-1}$ for all values of $\alpha \geq 0$ and $s \geq \alpha$, then this would show that $\mathrm{LEPT}_{\mathcal{F}}$ is an $(1+e^{-1})-$approximation algorithm (even for instances containing long jobs). We conjecture that this is true, which can hopefully be proved by analyzing the optimal value of Problem (8). Note that the objective function of this problem is convex, so $v^*(s,\alpha)$ must be reached at an extreme point of the polytope (8b)-(8d). We ran extensive numerical simulations that support our conjecture, but we did not obtain an analytical proof so far.

6 Performance of LEPT in the class of fixed assignment policies

It would also be interesting to characterize the approximation guarantee of $\mathrm{LEPT}_{\mathcal{F}}$ in the class of fixed assignment policies. The next proposition gives a lower bound:

Proposition 6.1. *For all $\epsilon > 0$, there exists an instance I of SEBP such that $\frac{\Phi(LEPT_{\mathcal{F}})}{OPT_{\mathcal{F}}(I)} = \frac{4-\epsilon}{3}$.*

Proof. We construct an instance with $m=2$ machines and $n=3$ jobs. The first two jobs are deterministic and have duration $P_1 = P_2 = 1$. The distribution of the third job is $P_3 = \frac{1}{\epsilon}X$, where $X \sim$ Bernoulli(ϵ), so $\mathbb{E}[P_3] = 1$. We assume that the $\mathrm{LEPT}_{\mathcal{F}}$ policy assigns both deterministic jobs to the first machine and the stochastic job to the other machine, which gives $\Phi(\mathrm{LEPT}_{\mathcal{F}}) = 2+(1-\epsilon)+\frac{\epsilon}{\epsilon} = 4-\epsilon$. In contrast, for any policy Π^* which assigns the two deterministic jobs on different machines, we have $\Phi(\Pi^*) = 1+(1-\epsilon)+(1+\frac{1}{\epsilon})\epsilon = 3$. The policy Π^* reaches the lower bound $m\max(\rho,1)$ of Proposition 2.1, hence it is optimal. □

If the conjecture in Section 5 is correct, this shows that the best approximation factor for $\text{LEPT}_{\mathcal{F}}$ in the class of fixed assignment policies lies between $\frac{4}{3}$ and $1 + e^{-1} \approx 1.368$.

7 Conclusion and Future work

We showed that $\text{LEPT}_{\mathcal{F}}$ is, in some sense, the best algorithm among the class of fixed assignment policies we can hope for. This result might inspire future work to consider the same or similar ratios for other scheduling problems, in which we compare within or against several subclasses of policies, in order to obtain more interesting and precise results on the performance of algorithms.

An interesting direction for future work on SEBP is the study of the case of unequal bins, which is relevant for the application to surgery scheduling, where operating rooms may have different opening hours. Since the class of fixed assignment policies is relevant for surgery scheduling, another interesting open question is whether there exists a policy $\Pi \in \mathcal{F}$ with an performance guarantee $< \frac{4}{3}$ in the class $\mathcal{F}$. A good candidate could be the variant of LEPT that considers more than just first moment information on the P_j's, and inserts sequentially the job j on the machine minimizing $\mathbb{E}[\max(X_i + P_j, 1)]$, where X_i is the random variable for the load already assigned to machine i. We also observe that the coefficient of variation of the jobs tend to infinity in all our tight examples, so it is natural to ask if we can obtain better bounds when these coefficients are upper bounded by a constant Δ. Last but not least, a two-stage stochastic online extension of the EBP could yield a better understanding of policies for the surgery scheduling problem with add-on cases (emergencies).

A Proofs of intermediate results

Proof of Proposition 2.2. To prove this result, we examine the change in the objective value of Π when we move one job to the machine with highest load in Π, for a realization $\boldsymbol{p}$ of the processing times. W.l.o.g. let machine 1 be the one with highest workload in $\Pi(\boldsymbol{p})$. Consider another machine $i \in \mathcal{M} \setminus \{1\}$ on which at least one job is scheduled. Let k be the last job on machine i, i.e., $C_k^{\Pi}(\boldsymbol{p}) = W_i^{\Pi}(\boldsymbol{p})$. For the sake of simplicity, we define $A := \{j \in \mathcal{J} | j \xrightarrow{\Pi(\boldsymbol{p})} i\} \setminus \{k\}$ and $B := \{j \in \mathcal{J} | j \xrightarrow{\Pi(\boldsymbol{p})} 1\}$. We consider another schedule $\Pi'(\boldsymbol{p})$ which coincides with $\Pi(\boldsymbol{p})$ except that job k is scheduled on machine 1 right after all jobs in B. We obtain

$$
\begin{aligned}
&\phi(\Pi, \boldsymbol{p}) - \phi(\Pi', \boldsymbol{p}) \\
=& \max\Big(\sum_{j\in A} p_j + p_k, 1\Big) + \max\Big(\sum_{j\in B} p_j, 1\Big) - \Big(\max\Big(\sum_{j\in A} p_j, 1\Big) + \max\Big(\sum_{j\in B} p_j + p_k, 1\Big)\Big) \\
=& \begin{cases} 1 + \max\Big(\sum_{j\in B} p_j, 1\Big) - \Big(1 + \max\Big(\sum_{j\in B} p_j + p_k, 1\Big)\Big) & \text{if } \sum_{j\in A} p_j + p_k \leq 1 \\ \sum_{j\in A} p_j + p_k + \sum_{j\in B} p_j - \Big(\max\Big(\sum_{j\in A} p_j, 1\Big) + \sum_{j\in B} p_j + p_k\Big) & \text{otherwise} \end{cases} \\
\leq& 0.
\end{aligned}
$$

Hence, iteratively moving some job k to the fullest machine yields $\phi(\Pi, \boldsymbol{p}) \leq \phi(\Pi_1, \boldsymbol{p})$. Finally, the result follows by taking the expectation. □

Proof of Lemma 2.5. The proof simply works by exploiting the analytical form of Poisson probabilities:

$$
\begin{aligned}
\frac{1}{\lambda}\mathbb{E}\Big[\max(Y, \lambda)\Big] &= \frac{1}{\lambda}\sum_{k=0}^{\infty} \max(k, \lambda) \cdot \frac{e^{-\lambda}\lambda^k}{k!} \\
&= \frac{1}{\lambda}\sum_{k=0}^{\infty} k \cdot \frac{e^{-\lambda}\lambda^k}{k!} + \frac{1}{\lambda}\sum_{k=0}^{\infty} \max(0, \lambda - k) \cdot \frac{e^{-\lambda}\lambda^k}{k!} \\
&= 1 + \sum_{k=0}^{\lambda}\Big(1 - \frac{k}{\lambda}\Big) \cdot \frac{e^{-\lambda}\lambda^k}{k!} \\
&= 1 + e^{-\lambda} \cdot \Big(\sum_{k=0}^{\lambda} \frac{\lambda^k}{k!} - \sum_{k=1}^{\lambda} \frac{\lambda^{k-1}}{(k-1)!}\Big) \\
&= 1 + \frac{e^{-\lambda}\lambda^{\lambda}}{\lambda!},
\end{aligned}
$$

where the last step follows from the property of a telescoping sum. □

Proof of Lemma 3.2. Let X and Y be random variables with $\mathbb{P}[0 \leq X \leq 1] = 1$. Observe that $0 \leq \mathbb{E}[X] \leq 1$. We are going to show that $\mathbb{E}[\max(X + Y, 1)]$ can be bounded from above by choosing the two point distribution $X^* \sim \text{Bernoulli}(\mathbb{E}[X])$, such that $\mathbb{P}[X^* = 0] = (1 - \mathbb{E}[X])$ and $\mathbb{P}[X^* = 1] = \mathbb{E}[X]$. To do so, we define the function $g : [0, 1] \to \mathbb{R}$, $x \mapsto \mathbb{E}_Y[\max(x + Y, 1)]$. This function is convex, since it is the expectation of a pointwise maximum of two affine functions [3]. Therefore, for all $x \in [0, 1]$ we have $g(x) \leq g(0) + x(g(1) - g(0))$. Then, by definition of g,

$$
\begin{aligned}
\mathbb{E}[\max(X + Y, 1)] = \mathbb{E}_X[g(X)] &\leq g(0) + \mathbb{E}_X[X] \cdot (g(1) - g(0)) \\
&= \mathbb{E}_{X^*}[g(X^*)] = \mathbb{E}[\max(X^* + Y, 1)].
\end{aligned}
$$

Using this bound for all $j \in [k]$, we obtain $\mathbb{E}\Big[\max\Big(\sum_{j=1}^{k} P_j, 1\Big)\Big] \leq \mathbb{E}\Big[\max\Big(\sum_{j=1}^{k} P_j^*, 1\Big)\Big]$, where $P_j^* \sim \text{Bernoulli}(\mathbb{E}[P_j])$. Then, by the law of total expectation, we have:

$$\mathbb{E}\Big[\max\Big(\sum_{j=1}^{k} P_j^*, 1\Big)\Big] = \mathbb{E}\Big[\sum_{j=1}^{k} P_j^* \Big| \sum_{j=1}^{k} P_j^* \geq 1\Big]\mathbb{P}[\sum_{j=1}^{k} P_j^* \geq 1] + \mathbb{E}[1]\ \mathbb{P}[\sum_{j=1}^{k} P_j^* < 1].$$

Since the random variable $\sum_{j=1}^{k} P_j^*$ is a nonnegative integer, it cannot lie in the interval $(0,1)$, so the first term in the above sum is equal to $\mathbb{E}\Big[\sum_{j=1}^{k} P_j^*\Big] = \sum_{j=1}^{k} \mathbb{E}\Big[P_j\Big]$, and the second term is equal to $\mathbb{P}[P_1^* = \ldots = P_k^* = 0] = \prod_{j=1}^{k}(1 - \mathbb{E}[P_j])$. □

Proof of Lemma 3.3. We set $\ell := \min\{x_i : i \in \mathcal{M}\}$. Then, the first inequality follows immediately. Next, we will show that in each step that $\text{LEPT}_{\mathcal{F}}$ assigns a job to a machine the second inequality is fulfilled. Let j denote the job which is put on machine i in the current step. Furthermore, let ℓ' and ℓ denote the minimum expected load among all machines before and after the allocation, respectively. Trivially, $\ell' \leq \ell$ is true. Moreover, let x_i' and x_i denote the expected workload of i before and after assigning j to it, respectively. Clearly, we have

$$x_i = x_i' + \mathbb{E}[P_j].$$

Observe, that $\ell' = x_i'$, because $\text{LEPT}_{\mathcal{F}}$ assigns j to the machine with the smallest expected load. In addition, let n_i denote the number of jobs running on machine i after the insertion of j. Since $\text{LEPT}_{\mathcal{F}}$ sorts jobs in decreasing order of their expected processing times, it holds

$$\mathbb{E}[P_j] \leq \frac{x_i'}{n_i - 1} = \frac{\ell'}{n_i - 1}.$$

Consider a machine other than i. If the inequality of the statement was fulfilled in an earlier step, then by setting the new ℓ it still is true. In the beginning, when we have no job at all, the inequality is true, so we only have to take care of machine i.
Finally, we obtain on machine i

$$\frac{x_i}{\ell} = \frac{x_i' + \mathbb{E}[P_j]}{\ell} \leq \frac{x_i' + \mathbb{E}[P_j]}{\ell'} \leq 1 + \frac{\mathbb{E}[P_j]}{\ell'} \leq 1 + \frac{\ell'}{\ell'(n_i - 1)} = \frac{n_i}{(n_i - 1)}.$$

□

Proof of Lemma 3.4. First, we argue that $h : y \mapsto (1-y)^{1+\frac{\ell}{y}}$ is convex over $[0,1]$. To see this, we compute its second derivative:

$$h''(y) = \frac{\ell(1-y)^{\frac{\ell}{y}-1}}{y^4} h_2(y),$$

where $h_2(y) := y^2(\ell - y + 2) + \ell(y-1)^2\log^2(1-y) - 2(\ell+1)(y-1)y\log(1-y)$. Now, we use the fact that $\log(1-y) = -\sum_{k=1}^{\infty} \frac{y^k}{k}$ for all $y \in [0,1)$. Hence, $\log^2(1-y) = \sum_{k=2}^{\infty} \gamma_k y^k$, where $\gamma_k := \sum_{i=1}^{k-1} \frac{1}{i(k-i)}$. After some calculus, the terms of order 2 and 3 vanish and we obtain the following series representation of h_2 over $[0,1)$:

$$h_2(y) = (\frac{\ell}{4} + \frac{1}{3})y^4 + \sum_{k=5}^{\infty}(\frac{2(\ell+1)}{(k-1)(k-2)} + \ell(\gamma_k + \gamma_{k-2} - 2\gamma_{k-1}))y^k.$$

We are going to show that $\gamma_k + \gamma_{k-2} - 2\gamma_{k-1} \geq 0$ for $k \geq 5$ implying that $h''(y) \geq 0$ for all $y \in [0, 1)$. To do so, we rewrite the sums using the partial fraction decomposition. As a consequence, we obtain

$$\begin{aligned}
\gamma_k + \gamma_{k-2} - 2\gamma_{k-1} &= \frac{2}{k}\sum_{i=1}^{k-1}\frac{1}{i} + \frac{2}{k-2}\sum_{i=1}^{k-3}\frac{1}{i} - \frac{4}{k-1}\sum_{i=1}^{k-2}\frac{1}{i} \\
&= \frac{2}{k}\left(\frac{1}{k-2} + \frac{1}{k-1}\right) - \frac{4}{(k-1)(k-2)} + \left(\frac{2}{k} + \frac{2}{k-2} - \frac{4}{k-1}\right)\sum_{i=1}^{k-3}\frac{1}{i} \\
&= -\frac{6}{k(k-1)(k-2)} + \frac{4}{k(k-1)(k-2)}\sum_{i=1}^{k-3}\frac{1}{i} \\
&\geq 0.
\end{aligned}$$

The last inequality results from the fact that for all $k \geq 5$ we have $4\sum_{i=1}^{k-3}\frac{1}{i} \geq 6$. Hence, h is convex on $[0, 1)$, and even on $[0, 1]$ by continuity. Now, let $v^*(\rho, \ell)$ denote the optimal value of the problem

$$\begin{aligned}
\underset{\boldsymbol{y}\in\mathbb{R}^m}{\textbf{maximize}} \quad & \sum_{i\in\mathcal{M}} h(y_i) && \text{(9a)} \\
s.t. \quad & \sum_{i\in\mathcal{M}} y_i = m(\rho - \ell) && \text{(9b)} \\
& 0 \leq y_i \leq 1, \quad (\forall i \in \mathcal{M}). && \text{(9c)}
\end{aligned}$$

As h is convex, a maximizer of the optimization problem above is an extreme point of the polytope induced by the constraints (9b) and (9c). Let $k := \lfloor m(\rho - \ell) \rfloor$ and $u := m(\rho - \ell) - k$, where $\lfloor . \rfloor$ denotes the floor function, that is, $\lfloor x \rfloor$ is the largest integer less than or equal to x. By construction, it holds $0 \leq u \leq 1$, and $u + k = m(\rho - \ell)$. At an extreme point, at least $m - 1$ inequalities of (9c) must be tight. Hence, one coordinate of $\boldsymbol{y}$ must be u, k coordinates must be 1 and the remaining $(m - k - 1)$ coordinates must be 0.

It follows that $v^*(\rho, \ell) = (m - k - 1)h(0) + h(u) = (m - k - 1)e^{-\ell} + (1 - u)^{1+\ell/u}$. Now, we observe that $(1 - u)^{\ell/u} \leq e^{-\ell}$, so

$$\begin{aligned}
& & (1-u)^{1+\ell/u} &\leq (1-u)e^{-\ell} \\
&\iff & (1-u)^{1+\ell/u} &\leq (1+k-m(\rho-\ell))e^{-\ell} \\
&\iff & \underbrace{(m-k-1)e^{-\ell} + (1-u)^{1+\ell/u}}_{v^*(\rho,\ell)} &\leq m(1-\rho+\ell)e^{-\ell},
\end{aligned}$$

where the first equivalence is due to the decomposition $m(\rho - \ell) = k + u$.

Finally, the inequality of the proposition follows from the fact that $(1 + \ell - \rho)e^{-\ell}$ is a nondecreasing function of ℓ over $[0, \rho]$. □

References

[1] N. Alon, Y. Azar, G.J. Woeginger, and T. Yadid. Approximation schemes for scheduling on parallel machines. *Journal of Scheduling*, 1(1):55–66, 1998.

[2] B.P. Berg and B.T. Denton. Fast approximation methods for online scheduling of outpatient procedure centers. *INFORMS Journal on Computing*, 29(4):631–644, 2017.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] R. Canetti and S. Irani. Bounding the power of preemption in randomized scheduling. *SIAM Journal on Computing*, 27(4):993–1015, 1998.

[5] J. R. Correa, M. Skutella, and J. Verschae. The power of preemption on unrelated machines and applications to scheduling orders. *Mathematics of Operations Research*, 37(2):379–398, 2012.

[6] P. Dell'Olmo, H. Kellerer, M.G. Speranza, and Z. Tuza. A 13/12 approximation algorithm for bin packing with extendable bins. *Information Processing Letters*, 65(5):229–233, 1998.

[7] P. Dell'Olmo and M.G. Speranza. Approximation algorithms for partitioning small items in unequal bins to minimize the total size. *Discrete Applied Mathematics*, 94(1-3):181–191, 1999.

[8] B.T. Denton, A.J. Miller, H.J. Balasubramanian, and T.R. Huschka. Optimal allocation of surgery blocks to operating rooms under uncertainty. *Operations research*, 58(4-part-1):802–816, 2010.

[9] F. Dexter and R.D. Traub. How to schedule elective surgical cases into specific operating rooms to maximize the efficiency of use of operating room time. *Anesthesia & Analgesia*, 94(4):933–942, 2002.

[10] M. R. Garey and D. S. Johnson. *Computers and intractability: a guide to NP-completeness*. WH Freeman and Company, San Francisco, 1979.

[11] S.G. Kolliopoulos and G. Steiner. Approximation algorithms for scheduling problems with a modified total weighted tardiness objective. *Operations research letters*, 35(5):685–692, 2007.

[12] M. Y. Kovalyov and F. Werner. Approximation schemes for scheduling jobs with common due date on parallel machines to minimize total tardiness. *Journal of Heuristics*, 8(4):415–428, 2002.

[13] M. Liu, Y. Xu, C. Chu, and F. Zheng. Online scheduling to minimize modified total tardiness with an availability constraint. *Theoretical Computer Science*, 410(47-49):5039–5046, 2009.

[14] A.W. Marshall, I. Olkin, and B.C. Arnold. *Inequalities: theory of majorization and its applications*. Elsevier, 1979.

[15] N. Megow, M. Uetz, and T. Vredeveld. Models and algorithms for stochastic online scheduling. *Mathematics of Operations Research*, 31(3):513–525, 2006.

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

[17] G. Sagnol, R. Borndörfer, M. Grima, M. Seeling, and C. Spies. Robust allocation of operating rooms with lognormal case durations. In *Proceedings of the 15th International Conference on Project Management and Scheduling (PMS 2016)*, pages 52–55, 2016.

[18] A. S. Schulz and M. Skutella. Scheduling unrelated machines by randomized rounding. *SIAM Journal on Discrete Mathematics*, 15(4):450–469, 2002.

[19] M. Skutella, M. Sviridenko, and M. Uetz. Unrelated machine scheduling with stochastic processing times. *Mathematics of operations research*, 41(3):851–864, 2016.

[20] A. J. Soper and V. A. Strusevich. Power of preemption on uniform parallel machines. 2014.

[21] M.G. Speranza and Z. Tuza. On-line approximation algorithms for scheduling tasks on identical machines with extendable working time. *Annals of Operations Research*, 86:491–506, 1999.