MATHEMATICS OF OPERATIONS RESEARCH
Vol. 31, No. 1, February 2006, pp. 85–108
ISSN 0364-765X | EISSN 1526-5471 | 06 | 3101 | 0085

DOI 10.1287/moor.1050.0170
© 2006 INFORMS

Average-Case and Smoothed Competitive Analysis of the Multilevel Feedback Algorithm

Luca Becchetti, Stefano Leonardi, Alberto Marchetti-Spaccamela
Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza," Via Salaria 113, 00198 Rome, Italy
{becchett@dis.uniroma1.it, leon@dis.uniroma1.it, alberto@dis.uniroma1.it}

Guido Schäfer
Institut für Mathematik, Technische Universität Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany,
schaefer@math.tu-berlin.de

Tjark Vredeveld
Department of Quantitative Economics, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands,
t.vredeveld@ke.unimaas.nl

In this paper, we introduce the notion of smoothed competitive analysis of online algorithms. Smoothed analysis has been proposed by Spielman and Teng [25] to explain the behavior of algorithms that work well in practice while performing very poorly from a worst-case analysis point of view. We apply this notion to analyze the multilevel feedback algorithm (MLF) to minimize the total flow time on a sequence of jobs released over time when the processing time of a job is only known at time of completion.

The initial processing times are integers in the range $[1, 2^K]$. We use a partial bit randomization model, i.e., the initial processing times are smoothed by changing the k least significant bits under a quite general class of probability distributions. We show that MLF admits a smoothed competitive ratio of $O((2^k/\sigma)^3 + (2^k/\sigma)^2 2^{K-k})$, where σ denotes the standard deviation of the distribution. In particular, we obtain a competitive ratio of $O(2^{K-k})$ if $\sigma = \Theta(2^k)$. We also prove an $\Omega(2^{K-k})$ lower bound for any deterministic algorithm that is run on processing times smoothed according to the partial bit randomization model. For various other smoothing models, including the additive symmetric smoothing one, which is a variant of the model used by Spielman and Teng [25], we give a higher lower bound of $\Omega(2^K)$.

A direct consequence of our result is also the first average-case analysis of MLF. We show a constant expected ratio of the total flow time of MLF to the optimum under several distributions including the uniform one.

Key words: average-case analysis; multilevel feedback; competitive analysis; scheduling
MSC2000 subject classification: Primary: 68W40; secondary: 60H99
OR/MS subject classification: Primary: Analysis of algorithms, suboptimal algorithms; secondary: probability, applications
History: Received April 7, 2004; revised March 1, 2005.

1. Introduction. *Smoothed analysis* was proposed by Spielman and Teng [25] as a hybrid between average-case and worst-case analysis to explain the success of algorithms that are known to work well in practice while presenting poor worst-case performance. The basic idea is to randomly perturb the initial input instances and to analyze the performance of the algorithm on the perturbed instances. The *smoothed complexity* of an algorithm as defined by Spielman and Teng [26] is the maximum over all input instances of the expected running time on the perturbed instances. Intuitively, the smoothed complexity of an algorithm is small if the worst-case instances are isolated in the (instance × running time) space. The striking result of Spielman and Teng [26] was to show that the smoothed complexity of the simplex method with a certain pivot rule and by perturbing the coefficients with a normal distribution is polynomial. In a series of later papers Blum and Dunagan [8], Spielman and Teng [24], Banderier et al. [2], and Beier and Vöcking [5, 6], smoothed analysis was successfully applied to characterize the time complexity of other problems.

Online algorithms, in contrast to traditional optimization techniques, have to make decisions without knowledge of the future. An online algorithm learns about a new piece of input data only at its release time. The standard yardstick for online algorithms has become *competitive analysis* (Sleator and Tarjan [23]), which measures the quality of an online algorithm by comparing its performance to that of an optimal offline algorithm that has full knowledge of the future. Competitive analysis often provides an overly pessimistic estimation of the performance of an algorithm, or fails to distinguish between algorithms that perform differently in practice, due to the presence of pathological bad instances that rarely occur. The analysis of online algorithms seems to be a natural field for the application of the idea of smoothed analysis.

Several attempts along the line of restricting the power of the adversary have already been taken in the past. A partial list of these efforts includes the access graph model to restrict the input sequences in online paging problems to specific patterns (Borodin et al. [10]) and the resource augmentation model for analyzing online scheduling algorithms (Kalyanasundaram and Pruhs [13]). More related to our work is the *diffuse adversary model* of Koutsoupias and Papadimitriou [15], a refinement of competitive analysis that assumes that the actual distribution of the input is chosen by a worst-case adversary out of a known class of possible distributions.

Smoothed competitive analysis. In this paper, we introduce the notion of *smoothed competitiveness*. The competitive ratio c of an online deterministic algorithm $\textsc{alg}$ for a cost minimization problem is defined as the supremum over all input instances of the ratio between the algorithm and the optimal cost, i.e., $c := \sup_{\check{I}}(\textsc{alg}(\check{I})/\textsc{opt}(\check{I}))$. Following the idea of Spielman and Teng [25], we smoothen the input instance according to some probability distribution f. Given an input instances $\check{I}$, we denote by $N(\check{I})$ the set of instances that are obtainable by smoothing the input instance $\check{I}$ according to f. We define the *smoothed competitive ratio* as

$$c := \sup_{\check{I}} \mathbf{E}_{I \overset{f}{\leftarrow} N(\check{I})} \left[\frac{\textsc{alg}(I)}{\textsc{opt}(I)} \right], \tag{1}$$

where the supremum is taken over all input instances $\check{I}$, and the expectation is taken according to f over all instances I in $N(\check{I})$.

We remark that defining the smoothed competitive ratio of $\textsc{alg}$ as the supremum over all instances $\check{I}$ of the ratio between the expected cost of the algorithm and the expected optimal cost would give an alternative and by all means reasonable notion of smoothed competitiveness. However, we are interested in analyzing the smoothed competitive ratio on a "per-instance basis," which we think gives a stronger notion of competitiveness, and therefore adopt the definition in (1); see also Scharbrodt et al. [21] for further comments on this.

This kind of analysis results in having the algorithm and the smoothing process together play a game against an adversary, in a way similar to the game played by a randomized online algorithm against its adversary. This definition of smoothed competitive ratio allows us to prove upper and lower bounds against different adversaries.

In a way similar to the analysis of randomized online algorithms (Borodin and El-Yaniv [9]), we define different types of adversaries. The *oblivious adversary* constructs the input sequence only on the basis of the knowledge of the algorithm and of the smoothing function f. We also define a stronger adversary, the *adaptive adversary*, that constructs the input instance revealed to the algorithm after time t also on the basis of the execution of the algorithm up to time t. This means that the choices of the adversary at some time t only depend on the state of the algorithm at time t. Both adversaries are charged with the optimal offline cost on the smoothed input instance. Considering the instance space, in the oblivious case $N(\check{I})$ is defined at the beginning, once the adversary has fixed $\check{I}$, while in the adaptive case $\check{I}$, and thus $N(\check{I})$, are themselves random variables, since they depend on the evolution of the algorithm.

Smoothed competitive analysis is substantially different from the diffuse adversary model. In this latter model, the probability distribution of the input instances is selected by a worst-case adversary, while in the model we use in this paper, the input instance is chosen by a worst-case adversary and later perturbed according to a specific distribution.

Multilevel feedback algorithm. One of the most successful online algorithms used in practice is the multilevel feedback algorithm (MLF) for processor scheduling in a time-sharing multitasking operating system. MLF is a *nonclairvoyant* scheduling algorithm, i.e., scheduling decisions are taken without knowledge of the time a job needs to be executed. Windows NT (Nutt [20]) and Unix (Tanenbaum [26]) have MLF at the very basis of their scheduling policies. The obvious goal is to provide a fast response to users. A widely used measure for the responsiveness of the system is the *average flow time* of the jobs, i.e., the average time spent by jobs in the system between release and completion. Job preemption is also widely recognized as a key factor to improve the responsiveness of the system. The basic idea of MLF is to organize jobs into a set of queues $Q_0, Q_1, \ldots$. Each job is processed for 2^i time units before being promoted to queue Q_{i+1} if not completed. At any time, MLF processes the job at the front of the lowest queue.

While MLF turns out to be very efficient in practice, it behaves poorly with respect to worst-case analysis. Motwani et al. [19] proved two lower bounds on the competitive ratio of any deterministic nonclairvoyant preemptive scheduling algorithm: an $\Omega(2^K)$ one, if the processing times are in $[1, 2^K]$, and an $\Omega(n^{1/3})$ one, where n is the number of released jobs. A randomized version of the multilevel feedback algorithm (RMLF) was first proposed by Kalyanasundaram and Pruhs [14] for a single machine achieving an $O(\log n \log\log n)$ competitive ratio against the online adaptive adversary. Becchetti and Leonardi [3] present a version of RMLF achieving an $O(\log n \log(n/m))$ competitive result on m parallel machines and a tight $O(\log n)$ competitive ratio on a single machine against the oblivious adversary, therefore matching for a single machine the randomized lower bound of Motwani et al. [19].

Our contribution. In this paper, we apply smoothed competitive analysis to the multilevel feedback algorithm. For smoothing the initial integral processing times, we use the *partial bit randomization* model. The idea is to replace the k least significant bits by some random number in $[1, 2^k]$. Our analysis holds for a wide class of distributions that we refer to as *well-shaped* distributions, including the uniform, the exponential symmetric, and the normal distribution. For k varying from 0 to K, we "smoothly" move from worst-case to average-case analysis.

(i) We show that MLF admits a smoothed competitive ratio of $O((2^k/\sigma)^3 + (2^k/\sigma)^2 2^{K-k})$, where σ denotes the standard deviation of the underlying distribution. The competitive ratio therefore improves exponentially with k and as the distribution becomes less sharply concentrated around its mean. In particular, if we smoothen according to the uniform distribution, we obtain an expected competitive ratio of $O(2^{K-k})$. We remark that our analysis holds for both the oblivious and the adaptive adversary. However, for the sake of clarity, we first concentrate on the oblivious adversary and discuss the differences for the adaptive adversary later.

As mentioned above, one could alternatively define the smoothed competitive ratio as the supremum over the set of possible input instances, of the ratio between the expected costs of the algorithm and the optimum. We point out that we obtain the same results under this alternative, weaker definition.

(ii) As a consequence of our analysis, we also obtain an average-case analysis of MLF. As an example, for $k = K$, our result implies an $O(1)$ expected ratio between the flow time of MLF and the optimum for all distributions with $\sigma = \Theta(2^k)$, therefore including the uniform one. Very surprisingly, to the best of our knowledge, this is the first average-case analysis of MLF.

Recently, Scharbrodt et al. [21] performed the analysis of the average competitive ratio of the shortest expected processing time first heuristic to minimize the average completion time where the processing times of the jobs follow a gamma distribution. Our result is stronger in the following aspects: (a) the analysis of Scharbrodt et al. [21] applies when the algorithm knows the distribution of the processing times, while in our analysis we require no knowledge about the distribution of the processing times, and (b) our result applies to average flow time, a measure of optimality stronger than average completion time. In an early work, Michel and Coffman [17] only considered the problem of synthesizing a feedback queue system under Poisson arrivals and a known discrete probability distribution on processing times so that prespecified mean flow time criteria are met.

(iii) We prove a lower bound of $\Omega(2^{K-k})$ against an adaptive adversary and a slightly weaker bound of $\Omega(2^{K/6-k/2})$, for every $k \le K/3$, against an oblivious adversary for any deterministic algorithm when run on processing times smoothed according to the partial bit randomization model.

(iv) Spielman and Teng [25] used an additive symmetric smoothing model, where each input parameter is smoothed symmetrically around its initial value. A natural question is whether a variant of this model is more suitable than the partial bit randomization model to analyze MLF. In fact, we prove that MLF admits a poor competitive ratio of $\Omega(2^K)$ under various other smoothing models, including the additive symmetric, the additive relative symmetric, and the multiplicative smoothing model.

(v) Our analysis holds if the processing times are smoothed by means of a partial bit randomization model. In many worst-case instances for this kind of scheduling problems, shortly before a job completes, the adversary releases a long job which delays the tiny fraction of the running job and thus its completion time. Hence, perturbing release dates slightly could weaken the adversary. A question that arises is whether smoothing the release dates additionally further reduces the smoothed competitive ratio of MLF. We answer this question in the negative by proving a lower bound of $\Omega(2^{K-k})$ on the smoothed competitive ratio of MLF if only the disruption of the release dates is not too large.

2. Problem definition and multilevel feedback algorithm.

The adversary releases a set $J := [n]$ of n jobs over time. For each job $j \in J$, the adversary specifies its *release time* r_j and its *initial processing time* $\check{p}_j$, which we assume to be an integer in $[1, 2^K]$. We consider the single machine case. The machine can process at most one job at a time and a job cannot be processed before its release time. We allow *preemption* of jobs, i.e., a job that is running can be interrupted and resumed later on the machine. A scheduling algorithm decides which uncompleted job should be executed at each time. For a generic schedule $\mathcal{S}$, let $C_j^{\mathcal{S}}$ denote the *completion time* of job j. The *flow time* of job j is given by $F_j^{\mathcal{S}} := C_j^{\mathcal{S}} - r_j$, i.e., the total time that j is in the system. The *total flow time* of a schedule $\mathcal{S}$ is defined as $F^{\mathcal{S}} := \sum_{j\in J} F_j^{\mathcal{S}}$ and the *average flow time* is given by $(1/n)F^{\mathcal{S}}$. A *nonclairvoyant* scheduling algorithm knows about the existence of a job only at the release time of the job, and the processing time of a job is only known when the job is completed. The objective is to find a schedule that minimizes the average flow time. In the clairvoyant case, i.e., when the algorithm knows the length of a job as soon as it is released, the problem is solved to optimality by the online algorithm shortest remaining processing time (SRPT) (Schrage [22]). This procedure schedules at any time the job which has least time left to be processed.

We review the multilevel feedback (MLF) algorithm. We say that a job is *alive* or *active* at time t in a generic schedule $\mathcal{S}$ if it has been released but not completed at this time, i.e., $r_j \le t < C_j^{\mathcal{S}}$. Denote by $x_j^{\mathcal{S}}(t)$ the amount of time that has been spent on processing job j in schedule $\mathcal{S}$ up to time t. We define $y_j^{\mathcal{S}}(t) := p_j - x_j^{\mathcal{S}}(t)$ as the *remaining processing time* of job j in schedule $\mathcal{S}$ at time t. In the sequel, we denote by MLF the schedule produced by the multilevel feedback algorithm.

The set of active jobs is partitioned into a set of priority queues $Q_0, Q_1, \ldots$. Within each queue, the priority is determined by the release dates of the jobs: The job with smallest release time has highest priority. For any two queues Q_h and Q_i, we say that Q_h is lower than Q_i if $h < i$. At any time t, MLF behaves as follows.

(i) Job j released at time t enters queue Q_0.
(ii) Schedule on the machine the alive job that has highest priority in the lowest nonempty queue.
(iii) For a job j in a queue Q_i at time t, if $x_j^{\text{MLF}}(t) = p_j$, assign $C_j^{\text{MLF}} := t$ and remove the job from the queue.
(iv) For a job j in a queue Q_i at time t, if $x_j^{\text{MLF}}(t) = 2^i < p_j$, job j is moved from Q_i to Q_{i+1}.

Observe that if the processing times are in $[1, 2^K]$, then at most $K+1$ queues $Q_0, \ldots, Q_K$ are used during the execution of MLF. Moreover, at any time t and for any queue Q_i, at most one job in Q_i has been executed. Put differently, if we consider all jobs that are in queue Q_i at time t, then at most one of these jobs satisfies $x_j^{\text{MLF}}(t) > 2^{i-1}$, while for all other jobs we have $x_j^{\text{MLF}}(t) = 2^{i-1}$.

FACT 1. *At any time t and for any queue Q_i, at most one job, alive at time t, has been executed in Q_i but has not been promoted to Q_{i+1}.*

Under which circumstances does MLF achieve a good performance guarantee? We offer some intuition. As mentioned, shortest remaining processing time (SRPT) is an optimal algorithm for the single machine case. We can view MLF as trying to simulate SRPT by using estimates for the processing times of the jobs in the system. When a new job arrives, its estimated processing time is 1; if a job is enqueued into queue Q_i, $i > 0$, MLF assumes that it has processing time 2^i. Put differently, whenever a job has been executed for its estimated processing time and is not completed, MLF doubles its estimate. Observe that if a job j is enqueued into queue Q_i, $i > 0$, MLF assumes that it takes 2^{i-1} additional time to complete j. Therefore, MLF gives precedence to jobs in lower queues.

Consider a job j with processing time $p_j \in (2^{i-1}, 2^i]$. The final estimate of j's processing time in MLF is 2^i. Intuitively, if the actual processing time of j is not too far from its final estimate, then the decisions made by MLF with respect to j are not too different from those made by SRPT. However, if the final estimate is far off from the actual processing time, then MLF and SRPT may indeed perform very differently. For example, suppose that the actual processing time of j is $2^{i-1}+1$. When j enters queue Q_i, MLF defers j until all jobs of processing time at most 2^{i-1} are completed. On the other hand, SRPT completes j after one additional time unit.

In fact, it can easily be seen that MLF may perform arbitrarily bad on jobs of the latter kind: We release jobs in two phases. In the first phase, at time $t = 0$, we release $N := 2^{K-1}+1$ jobs with processing time $2^{K-1}+1$. Let $\hat{t}$ be the first time when a job, say j^*, has been completed by MLF. At time $\hat{t}$, all remaining $N-1$ jobs have remaining processing time 1. Now, consider another algorithm ALG that does not schedule j^*, and therefore can allocate $2^{K-1}+1$ time units on the other jobs. ALG will have completed all jobs except j^* by time $\hat{t}$. In the second phase, starting at time $\hat{t}$, we release one after another a long sequence of jobs with processing time 1. If we choose this sequence sufficiently long, the total flow time will be dominated by the contribution of the second phase. Since, during the second phase, MLF has at least N jobs in the system while ALG has only two jobs in the system, we obtain a competitive ratio of $\Omega(N) = \Omega(2^K)$.

3. Smoothing models.

We smoothen the processing times of the jobs. We remark that we could additionally smoothen the release dates. However, for our analysis to hold, it is sufficient to only smoothen the processing times. Furthermore, from a practical point of view, each job is released at a certain time, while processing times are estimates. Therefore, it is more natural to smoothen the processing times and to leave the release dates intact. As will be seen in §6, smoothing the release dates additionally does not further improve the smoothed competitive ratio of MLF.

The input instance may be smoothed according to different smoothing models. We discuss four different smoothing models below.

Additive symmetric smoothing model. In the additive symmetric smoothing model, the processing time of each job is smoothed symmetrically around its initial processing time. The difference between the smoothed and original processing time $p_j - \check{p}_j$ of a job j is drawn independently at random according to some probability function f from a range $[-L, L]$, for some L. Here, L is the same for all processing times (a similar model is used by Spielman and Teng [25]):

$$p_j := \max(1, \check{p}_j + \varepsilon_j), \quad \text{where } \varepsilon_j \xleftarrow{f} [-L, L].$$

The maximum is taken in order to assure that the smoothed processing times are at least 1.

Additive relative symmetric smoothing model. The additive relative symmetric smoothing model is similar to the previous one. Here, however, the range of the smoothed processing time of j depends on its initial processing time $\check{p}_j$. More precisely, for $c < 1$, the smoothed processing time p_j of j is defined as

$$p_j := \max(1, \check{p}_j + \varepsilon_j), \quad \text{where } \varepsilon_j \overset{f}{\leftarrow} [-(\check{p}_j)^c, (\check{p}_j)^c].$$

Multiplicative smoothing model. In the multiplicative smoothing model, the processing time of each job is smoothed symmetrically around its initial processing time. The smoothed processing times are chosen independently according to f from the range $[(1-\epsilon)\check{p}_j, (1+\epsilon)\check{p}_j]$ for some $\epsilon > 0$. This model is also discussed but not analyzed by Spielman and Teng [25]:

$$p_j := \max(1, \check{p}_j + \varepsilon_j), \quad \text{where } \varepsilon_j \overset{f}{\leftarrow} [-\epsilon\check{p}_j, \epsilon\check{p}_j].$$

Partial bit randomization model. The initial processing times are smoothed by changing the k least significant bits at random according to some probability function f. More precisely, the smoothed processing time p_j of a job j is defined as

$$p_j := 2^k \left\lfloor \frac{\check{p}_j - 1}{2^k} \right\rfloor + \varepsilon_j, \quad \text{where } \varepsilon_j \overset{f}{\leftarrow} [1, 2^k].$$

Note that ε_j is at least 1 and therefore 1 is subtracted from $\check{p}_j$ before applying the modification. For $k = 0$, the smoothed processing times are equal to the initial processing times; for $k = K$, the processing times are chosen entirely at random from $[1, 2^K]$.

We will show that MLF has a smoothed competitive ratio of $\Omega(2^K)$ under the first three smoothing models. Therefore, these models are not suitable to explain the success of MLF in practice. The model we use is the partial bit randomization model. A similar model is used by Beier et al. [7] and by Banderier et al. [2]. However, in Beier et al. [7] and Banderier et al. [2], only the uniform distribution was considered, while our analysis holds for a large class of smoothing distributions. At first glance, it may seem odd to allow distributions other than the uniform one. However, the advantage is that for $k = K$ we obtain processing times that are chosen entirely at random according to f.

3.1. Feasible smoothing distributions.

Our analysis holds for any smoothing distribution f that satisfies properties (P1), (P2), and (P3) below. Let ε be a random variable that is chosen according to density function f from $[1, 2^k]$.

(P1) $\mathbf{P}\left[\varepsilon \geq (1+\gamma)2^{k-1}\right] \geq \alpha$ for some $0 < \alpha \leq 1$ and $0 < \gamma \leq 2^{k-K-1}$.

(P2) $\sum_{i=0}^{k} \mathbf{P}\left[\varepsilon \leq 2^i\right] \leq \beta$ for some $1 \leq \beta \leq k+1$.

(P3) $\mathbf{E}[\varepsilon] \geq \delta \cdot 2^k$ for some $0 < \delta \leq 1$.

We provide some intuition; see also Figure 1. (P1) states that the upper tail probability of f is at least α. Supposing β is small, (P2) means that f is slowly increasing from 1. (P3) states that the expectation of f is not too close to 1. Note that (P2) and (P3) can be trivially satisfied by choosing $\beta = k+1$ and $\delta = 1/2^k$. However, constant values for β and δ improve the smoothed competitive ratio. We remark that our analysis holds for both discrete and continuous distributions. In the sequel, however, we assume that f is discrete. We use μ and σ to denote the expectation and standard deviation of f, respectively.

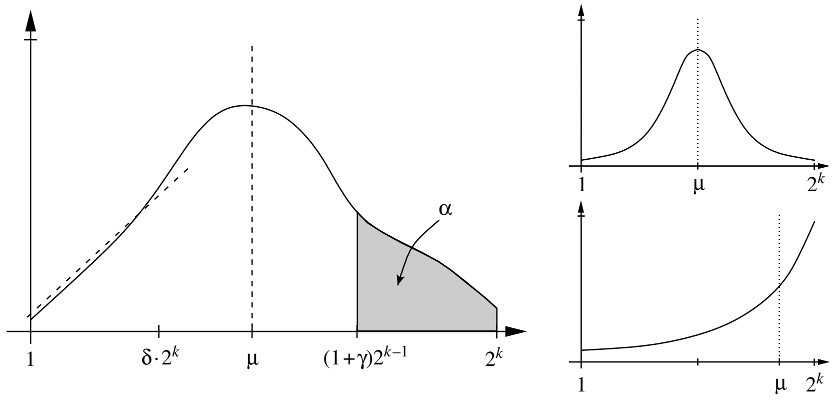

FIGURE 1. Illustration of properties (P1)–(P3).

For distributions satisfying (P1)–(P3), we prove that MLF has smoothed competitive ratio

$$O\left(\frac{K-k+\beta}{\alpha}+\frac{1}{\alpha\gamma}+\frac{1}{\delta^2}\right).$$

Ideally, if α, β, and δ are constants and $\gamma = 2^{k-K-1}$, we obtain a smoothed competitive ratio of $O(2^{K-k})$. It is difficult to give a generic characterization for distributions that satisfy (P1)–(P3) with reasonable values α, γ, β, and δ. We propose the following class of distributions and refer the reader to Appendix B for further characterizations. We call a distribution f *well shaped* if the following conditions hold:

(i) f is symmetric around μ,
(ii) $\mu \geq 2^{k-1}$, and
(iii) f is nondecreasing in $[1, 2^{k-1}]$.

For example, the uniform, the normal, and the double exponential distribution with $\mu = 2^{k-1}+\frac{1}{2}$ are well-shaped distributions. In Appendix B, we show that well-shaped distributions satisfy (P1)–(P3) with

$$\alpha = \left(\frac{\sigma}{2^k}\right)^2, \quad \gamma = \min\left(\frac{1}{\sqrt{2}}\left(\frac{\sigma}{2^{k-1}}\right), 2^{k-K-1}\right), \quad \beta = 2, \quad \text{and} \quad \delta = \frac{1}{2}.$$

Therefore, for a well-shaped distribution we obtain a smoothed competitive ratio of

$$O\left(\left(\frac{2^k}{\sigma}\right)^3+\left(\frac{2^k}{\sigma}\right)^2 2^{K-k}\right).$$

From the discussion in Appendix B, it will also become apparent that we obtain the same competitive ratio for any distribution with $\mu \geq 2^{k-1}$ and which is nondecreasing in $[1, 2^k]$, e.g., for the exponential distribution.

3.2. Properties of smoothed processing times. We state two crucial properties of smoothed processing times. Define $\phi_j := 2^k\lfloor(\check{p}_j - 1)/2^k\rfloor$. We have $p_j = \phi_j + \varepsilon_j$. Consider a job j with initial processing time $\check{p}_j \in [1, 2^k]$. Then, the initial processing time of j is entirely replaced by some random processing time in $[1, 2^k]$ that is chosen according to the probability distribution f.

Fact 2. *For each job j with $\check{p}_j \in [1, 2^k]$ we have $\phi_j = 0$ and thus $p_j \in [1, 2^k]$. Moreover,* $\mathbf{P}[p_j \leq x] = \mathbf{P}[\varepsilon_j \leq x]$ *for each $x \in [1, 2^k]$.*

Next, consider a job j with initial processing time $\check{p}_j \in (2^{i-1}, 2^i]$ for some integer $i > k$. Then, the smoothed processing time p_j is randomly chosen from a subrange of $(2^{i-1}, 2^i]$ according to the probability distribution f.

Fact 3. *For each job j with $\check{p}_j \in (2^{i-1}, 2^i]$, for some integer $i > k$, we have $\phi_j \in [2^{i-1}, 2^i - 2^k]$ and thus $p_j \in (2^{i-1}, 2^i]$.*

4. Preliminaries. We use MLF and OPT to denote the schedules produced by the multilevel feedback algorithm and by an optimal algorithm, respectively. We use $\mathcal{S}$ to refer to a generic schedule.

We partition jobs into classes: A job $j \in J$ is of *class* $i \in [0, K]$ if $p_j \in (2^{i-1}, 2^i]$. We use $Cl_j \in [0, K]$ to denote the class of a job j. Note that, for $\check{p}_j \in (2^{i-1}, 2^i]$, with $i > k$, Cl_j is *not* a random variable; see Fact 3. In MLF, a job of class i will be completed in queue Q_i.

We denote by $\delta^{\mathcal{S}}(t)$ the number of jobs that are active at time t in $\mathcal{S}$. For each job j and any time t we define a binary random variable $X_j^{\mathcal{S}}(t)$: $X_j^{\mathcal{S}}(t)$ is 1 if job j is active at time t, and 0 otherwise. We have $\delta^{\mathcal{S}}(t) = \sum_{j\in J} X_j^{\mathcal{S}}(t)$. Moreover, we use $S^{\mathcal{S}}(t)$ to refer to the set of active jobs at time t.

The total flow time $F^{\mathcal{S}}$ of a schedule $\mathcal{S}$ is defined as the sum of the flow times of all jobs. Equivalently, we can express the total flow time as the integral over time of the number of active jobs. We state this as a fact; see also Leonardi and Raz [16].

Fact 4. $F^{\mathcal{S}} = \sum_{j\in J} F_j^{\mathcal{S}} = \int_{t\geq 0} \delta^{\mathcal{S}}(t)\,dt$.

The following obvious fact states that the sum of the processing times of all jobs is a lower bound on the flow time of any schedule $\mathcal{S}$.

Fact 5. $F^{\mathcal{S}} \geq \sum_{j\in J} p_j$.

An important notion in our analysis is the notion of *lucky* and *unlucky* jobs. It serves to distinguish between jobs that are good and those which are bad for the performance of MLF.

Definition 4.1. A job j of class i is called *lucky* if $p_j \geq (1+\gamma)2^{i-1}$; otherwise, it is called *unlucky*.

For each job j, we define a binary random variable X_j^l which is 1 if j is lucky, and 0 otherwise. Note that for MLF a lucky job of class i is a job that still has a remaining processing time of at least $\gamma 2^{i-1}$ when it enters its queue Q_i of completion. We use $\delta^l(t)$ to denote the number of lucky jobs that are active at time t in MLF. We also define a binary random variable $X_j^l(t)$ that indicates whether or not a job j is lucky and alive at time t in MLF, i.e., $X_j^l(t) := X_j^l \cdot X_j^{\text{MLF}}(t)$. We have $\delta^l(t) = \sum_{j\in J} X_j^l(t)$.

At time t, the job with highest priority among all jobs in queue Q_i (if any) is said to be the *head* of Q_i. A head job of queue Q_i is *ending* if it will be completed in Q_i. We denote by $h(t)$ the total number of ending head jobs at time t.

Let X be a generic random variable. For an input instance I, X_I denotes the value of X for this particular instance I. Note that X_I is uniquely determined by the execution of the algorithm.

5. Smoothed competitive analysis of MLF.

The intuition behind our analysis is as follows. We argued that MLF tries to simulate SRPT by using estimates of the processing times and that the performance of MLF can be related to the one of SRPT if the final estimates are not too far from the actual processing times of the jobs. We make this relation explicit by proving that, at any time t, the number of lucky jobs is at most the number of ending head jobs plus $6/\gamma$ times the number of active jobs in an optimal schedule. This argument is purely deterministic. We also prove an upper bound of $K-k+\beta$ on the expected number of ending head jobs at any time t.

We write the total flow time as the integral over time of the number of active jobs. At any time t, we distinguish between (i) the number of active jobs in MLF is at most $2/\alpha$ times the number of lucky jobs, and (ii) where this is not the case. We prove that case (i) occurs with high probability so that we can use the deterministic bound to relate MLF to the optimal algorithm. The contribution of case (ii) is compensated by the exponentially small probability of its occurrence.

The high-probability argument is presented in §5.1. Our analysis holds both for the oblivious adversary and for the adaptive adversary. For the sake of clarity, we first concentrate on the oblivious adversary and discuss the differences for the adaptive adversary in §5.2.

Lemma 5.1 provides a deterministic bound on the number of lucky jobs in the schedule of MLF for a specific instance I. The proof is similar to the one given by Becchetti and Leonardi [3] and can be found in Appendix C.

Lemma 5.1. *For any input instance I, at any time t, $\delta_I^l(t) \leq h_I(t) + (6/\gamma)\delta_I^{\text{OPT}}(t)$.*

Clearly, at any time t, the number of ending head jobs is at most $K+1$. The following lemma gives a better upper bound on the expected number of ending head jobs.

Lemma 5.2. *At any time t, $\mathbf{E}[h(t)] \leq K-k+\beta$.*

Proof. Let $h'(t)$ denote the number of ending head jobs in the first $k+1$ queues. Clearly $\mathbf{E}[h(t)] \leq K-k+\mathbf{E}[h'(t)]$, since the last $K-k$ queues can contribute at most $K-k$ to the expected value of $h(t)$.

We next consider the expected value of $h'(t)$. Let $H(t)$ denote the ordered sequence $(q_0, \ldots, q_k)$ of jobs that are at time t at the head of the first $k+1$ queues $Q_0, \ldots, Q_k$, respectively. We use $q_i = \times$ to denote that Q_i is empty at time t. Let $H_i(t)$ be a binary random variable indicating whether or not the head job of queue Q_i (if any) is ending, i.e., $H_i(t) = 1$ if $q_i \neq \times$ and q_i is in its final queue, and $H_i(t) = 0$ otherwise. Let $H \in (J \cup \times)^k$ denote any possible configuration for $H(t)$. Observe that by definition $\mathbf{P}[H_i(t)=1 \mid H(t)=H] = 0$ if $q_i = \times$. Let $q_i \neq \times$. We have

$$\mathbf{P}[H_i(t)=1 \mid H(t)=H] = \mathbf{P}\left[p_{q_i} \leq 2^i \mid H(t)=H\right].$$

In Appendix D, we show that the events $(p_{q_i} \leq 2^i)$ and $(H(t)=H)$ are negatively correlated. Thus, $\mathbf{P}[H_i(t)=1 \mid H(t)=H] \leq \mathbf{P}\left[p_{q_i} \leq 2^i\right]$. We obtain

$$\mathbf{E}[h'(t) \mid H(t)=H] = \sum_{i=0}^{k} \mathbf{P}[H_i(t)=1 \mid H(t)=H] \leq \sum_{i=0}^{k} \mathbf{P}\left[p_{q_i} \leq 2^i\right].$$

If a job q_i is of class larger than k, we have $\mathbf{P}\left[p_{q_i} \leq 2^i\right] = 0$. Therefore, the sum is maximized if we assume that each q_i is of class at most k. Since the processing times are chosen identically, independently, and (under the above assumption) entirely at random, we have

$$\mathbf{E}[h'(t) \mid H(t)=H] \leq \sum_{i=0}^{k} \mathbf{P}\left[\varepsilon_{q_i} \leq 2^i\right] \leq \sum_{i=0}^{k} \mathbf{P}\left[\varepsilon \leq 2^i\right] \leq \beta,$$

where ε is a random variable chosen according to f from $[1, 2^k]$, and the last inequality follows from property (P2) of our distribution. We conclude

$$\mathbf{E}[h'(t)] = \sum_{H \in (J \cup \times)^k} \mathbf{E}[h'(t) \mid H(t) = H]\,\mathbf{P}[H(t) = H] \le \beta. \quad \square$$

We define a random variable R as the sum of the random parts of all processing times, i.e., $R := \sum_{j \in J} \varepsilon_j$. We need the following bound on the probability that R is at least a constant fraction of its expectation.

Lemma 5.3. $\mathbf{P}\left[R \ge \frac{1}{2}\mathbf{E}[R]\right] \ge 1 - \mathrm{e}^{-n\delta^2/2}$.

Proof. Observe that $\mathbf{E}[R] = n\mu$, where μ denotes the expectation of f. We use Hoeffding's bound (Appendix A, Theorem A.5) and property (P3) to obtain

$$\mathbf{P}\left[R \le \frac{1}{2}\mathbf{E}[R]\right] \le \exp\left(-\frac{(1/2)\mathbf{E}[R]^2}{n(2^k-1)^2}\right) \le \exp\left(-\frac{(1/2)n\mu^2}{2^{2k}}\right) \le \exp\left(-\frac{n\delta^2}{2}\right). \quad \square$$

We are now in a position to prove Theorem 5.1. We introduce the following notation. For an instance I, we define

$$\mathscr{D}_I := \left\{t : \delta_I^{\text{MLF}}(t) \le \frac{2}{\alpha}\delta_I^l(t)\right\} \quad \text{and} \quad \overline{\mathscr{D}}_I := \left\{t : \delta_I^{\text{MLF}}(t) > \frac{2}{\alpha}\delta_I^l(t)\right\}.$$

Moreover, we define the event $\mathscr{E} := (R \ge \frac{1}{2}\mathbf{E}[R])$ and use $\overline{\mathscr{E}}$ to refer to the complement of $\mathscr{E}$.

Theorem 5.1. *For any instance $\check{I}$ and any smoothing distribution f that satisfies* (*P*1)–(*P*3),

$$\mathbf{E}_{I \xleftarrow{f} N(\check{I})}\left[\frac{F_I^{\text{MLF}}}{F_I^{\text{OPT}}}\right] = O\left(\frac{K-k+\beta}{\alpha} + \frac{1}{\alpha\gamma} + \frac{1}{\delta^2}\right).$$

Proof. For notational convenience, we omit the subscripts "I" and "$I \xleftarrow{f} N(\mathring{F})$" throughout this proof.

$$\mathbf{E}\left[\frac{F^{\text{MLF}}}{F^{\text{OPT}}}\right] = \mathbf{E}\left[\frac{F^{\text{MLF}}}{F^{\text{OPT}}}\,\middle|\,\mathscr{E}\right]\mathbf{P}[\mathscr{E}] + \mathbf{E}\left[\frac{F^{\text{MLF}}}{F^{\text{OPT}}}\,\middle|\,\overline{\mathscr{E}}\right]\mathbf{P}[\overline{\mathscr{E}}] \le \mathbf{E}\left[\frac{F^{\text{MLF}}}{F^{\text{OPT}}}\,\middle|\,\mathscr{E}\right]\mathbf{P}[\mathscr{E}] + n\mathrm{e}^{-n\delta^2/2},$$

where the inequality follows from Lemma 5.3 and the fact that n is an upper bound on the competitive ratio of MLF. Define $c := 2/\delta^2$. Since $\mathrm{e}^{-x} < 1/x$ for $x > 0$, we have $n\mathrm{e}^{-n\delta^2/2} < c$. We partition the flow time $F^{\text{MLF}} = \int_t \delta^{\text{MLF}}(t)\,dt$ into the contribution of time instants $t \in \mathscr{D}$ and $t \in \overline{\mathscr{D}}$, i.e., $F^{\text{MLF}} = \int_{t\in\mathscr{D}} \delta^{\text{MLF}}(t)\,dt + \int_{t\in\overline{\mathscr{D}}} \delta^{\text{MLF}}(t)\,dt$, and bound these contributions separately. More precisely:

$$\mathbf{E}\left[\frac{F^{\text{MLF}}}{F^{\text{OPT}}}\,\middle|\,\mathscr{E}\right] = \mathbf{E}\left[\frac{\int_{t\in\mathscr{D}} \delta^{\text{MLF}}(t)\,dt}{F^{\text{OPT}}}\,\middle|\,\mathscr{E}\right] + \mathbf{E}\left[\frac{\int_{t\in\overline{\mathscr{D}}} \delta^{\text{MLF}}(t)\,dt}{F^{\text{OPT}}}\,\middle|\,\mathscr{E}\right].$$

Now,

$$\begin{aligned}
\mathbf{E}\left[\frac{\int_{t\in\mathscr{D}} \delta^{\text{MLF}}(t)\,dt}{F^{\text{OPT}}}\,\middle|\,\mathscr{E}\right]\mathbf{P}[\mathscr{E}] &\le \mathbf{E}\left[\frac{\int_{t\in\mathscr{D}} (2/\alpha)\delta^l(t)\,dt}{F^{\text{OPT}}}\,\middle|\,\mathscr{E}\right]\mathbf{P}[\mathscr{E}] \\
&\le \mathbf{E}\left[\frac{\int_{t\in\mathscr{D}} (2/\alpha)h(t)\,dt + \int_{t\in\mathscr{D}} (2/\alpha)\cdot(6/\gamma)\delta^{\text{OPT}}(t)\,dt}{F^{\text{OPT}}}\,\middle|\,\mathscr{E}\right]\mathbf{P}[\mathscr{E}] \\
&\le \mathbf{E}\left[\frac{\int_{t\in\mathscr{D}} (2/\alpha)h(t)\,dt}{F^{\text{OPT}}}\,\middle|\,\mathscr{E}\right]\mathbf{P}[\mathscr{E}] + \frac{12}{\alpha\gamma},
\end{aligned}$$

where we use the deterministic bound of Lemma 5.1 on $\delta^l(t)$ and the fact that $F^{\text{OPT}} \ge \int_{t\in\mathscr{D}} \delta^{\text{OPT}}(t)\,dt$. By Fact 5 and the definition of event $\mathscr{E}$, we have $F^{\text{OPT}} \ge \sum_j p_j \ge \sum_j \phi_j + \frac{1}{2}\mathbf{E}[R]$. Hence,

$$\begin{aligned}
\mathbf{E}\left[\frac{\int_{t\in\mathscr{D}} \delta^{\text{MLF}}(t)\,dt}{F^{\text{OPT}}}\,\middle|\,\mathscr{E}\right]\mathbf{P}[\mathscr{E}] &\le \frac{\mathbf{E}\left[\int_{t\in\mathscr{D}} (2/\alpha)h(t)\,dt \mid \mathscr{E}\right]\mathbf{P}[\mathscr{E}]}{\sum_j \phi_j + (1/2)\mathbf{E}[R]} + \frac{12}{\alpha\gamma} \\
&\le \frac{(2/\alpha)(K-k+\beta)\mathbf{E}[\sum_j p_j]}{\sum_j \phi_j + (1/2)\mathbf{E}[R]} + \frac{12}{\alpha\gamma},
\end{aligned}$$

where we use Lemma 5.2 together with the fact that for any input instance $h(t)$ contributes only in those time instants where at least one job is in the system, so at most $\sum_j p_j$. Since $\mathbf{E}[\sum_j p_j] = \sum_j \phi_j + \mathbf{E}[R]$, we obtain

$$\mathbf{E}\left[\frac{\int_{t\in\mathscr{D}} \delta^{\text{MLF}}(t)\,dt}{F^{\text{OPT}}}\,\middle|\,\mathscr{E}\right]\mathbf{P}[\mathscr{E}] \le \frac{4(K-k+\beta)}{\alpha} + \frac{12}{\alpha\gamma}.$$

Next, consider the contribution of time instants $t \in \bar{\mathscr{D}}$. Given $\mathscr{E}$, we have $F^{\text{OPT}} \ge \sum_j \phi_j + \frac{1}{2}\mathbf{E}[R]$. Exploiting Lemma 5.4, which is given below, we obtain

$$\mathbf{E}\left[\frac{\int_{t\in\bar{\mathscr{D}}} \delta^{\text{MLF}}(t)\,dt}{F^{\text{OPT}}}\,\middle|\,\mathscr{E}\right]\mathbf{P}[\mathscr{E}] \le \frac{\mathbf{E}[\int_{t\in\bar{\mathscr{D}}} \delta^{\text{MLF}}(t)\,dt \mid \mathscr{E}]\mathbf{P}[\mathscr{E}]}{\sum_j \phi_j + (1/2)\mathbf{E}[R]} \le \frac{(8/\alpha)\,\mathbf{E}[\sum_j p_j]}{\sum_j \phi_j + (1/2)\mathbf{E}[R]} \le \frac{16}{\alpha}.$$

Putting everything together, we obtain

$$\mathbf{E}\left[\frac{F^{\text{MLF}}}{F^{\text{OPT}}}\right] \le \frac{4(K-k+\beta)}{\alpha} + \frac{12}{\alpha\gamma} + \frac{16}{\alpha} + \frac{2}{\delta^2}. \quad \square$$

Lemma 5.4. $\mathbf{E}\big[\int_{t\in\bar{\mathscr{D}}} \delta^{\text{MLF}}(t)\,dt \mid \mathscr{E}\big]\mathbf{P}[\mathscr{E}] \le (8/\alpha)\mathbf{E}[\sum_j p_j]$.

Proof. We use Lemma 5.5, the proof of which is the subject of §5.1. We have

$$\begin{aligned}
\mathbf{E}\left[\int_{t\in\bar{\mathscr{D}}} \delta^{\text{MLF}}(t)\,dt\,\middle|\,\mathscr{E}\right]\mathbf{P}[\mathscr{E}] &\le \mathbf{E}\left[\int_{t\in\bar{\mathscr{D}}} \delta^{\text{MLF}}(t)\,dt\right] \\
&= \int_{t\ge 0} \mathbf{E}\big[\delta^{\text{MLF}}(t) \mid t \in \bar{\mathscr{D}}\big]\mathbf{P}\big[t \in \bar{\mathscr{D}}\big]\,dt \\
&= \int_{t\ge 0} \sum_{s=1}^{n} s\,\mathbf{P}\big[\delta^{\text{MLF}}(t) = s \mid t \in \bar{\mathscr{D}}\big]\mathbf{P}\big[t \in \bar{\mathscr{D}}\big]\,dt \\
&= \int_{t\ge 0} \sum_{s=1}^{n} s\,\mathbf{P}\big[t \in \bar{\mathscr{D}} \mid \delta^{\text{MLF}}(t) = s\big]\mathbf{P}[\delta^{\text{MLF}}(t) = s]\,dt \\
&\le \int_{t\ge 0} \sum_{s=1}^{n} s\,\mathrm{e}^{-\alpha s/8}\mathbf{P}[\delta^{\text{MLF}}(t) = s]\,dt \\
&\le \frac{8}{\alpha}\int_{t\ge 0} \sum_{s=1}^{n} \mathbf{P}[\delta^{\text{MLF}}(t) = s]\,dt \\
&= \frac{8}{\alpha}\int_{t\ge 0} \mathbf{P}[\delta^{\text{MLF}}(t) \ge 1]\,dt \\
&= \frac{8}{\alpha}\mathbf{E}[\textstyle\sum_j p_j],
\end{aligned}$$

where the fifth inequality is due to Lemma 5.5 and the sixth inequality follows since $\mathrm{e}^{-x} < 1/x$ for $x > 0$. $\square$

5.1. High probability bound. To complete the proof, we are left to show that with high probability at any time t, the number of lucky jobs is a good fraction of the overall number of jobs in the system. This is stated by the following lemma:

Lemma 5.5. *For any $s \in [n]$, at any time t,* $\mathbf{P}\big[\delta^{l}(t) < \frac{1}{2}\alpha\delta^{\text{MLF}}(t) \mid \delta^{\text{MLF}}(t) = s\big] \le \mathrm{e}^{-\alpha s/8}$.

We first give a high-level idea of the proof of Lemma 5.5. Let $S \subseteq J$. We condition the probability space on the event that (i) the set of jobs that are alive at time t in MLF is equal to S, i.e., $(S^{\text{MLF}}(t) = S)$, and (ii) the processing times of all jobs not in S are fixed to values that are specified by a vector $\mathbf{x}_{\bar{S}}$, which we denote by $(\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}})$. We define the event $\mathscr{F}(t, S, \mathbf{x}_{\bar{S}}) := ((S^{\text{MLF}}(t) = S) \cap (\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}}))$.

Recall that we defined $X_j^l(t) = X_j^l \cdot X_j^{\text{MLF}}(t)$. Since we condition on $(S^{\text{MLF}}(t) = S)$, we have for each $j \in J$

$$X_j^l(t) = \begin{cases} X_j^l & \text{if } j \in S, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

Thus,

$$\mathbf{E}[\delta^l(t)\mid\mathcal{F}(t,S,\mathbf{x}_{\bar S})]=\sum_{j\in J}\mathbf{P}\left[X_j^l(t)=1\mid\mathcal{F}(t,S,\mathbf{x}_{\bar S})\right]=\sum_{j\in S}\mathbf{P}\left[X_j^l=1\mid\mathcal{F}(t,S,\mathbf{x}_{\bar S})\right].$$

In order to prove Lemma 5.5, we proceed as follows. We first prove that, conditioned on $\mathcal{F}(t,S,\mathbf{x}_{\bar S})$, the random variables $(X_j^l\mid\mathcal{F}(t,S,\mathbf{x}_{\bar S}))$, $j\in S$, are independent (Lemma 5.8). After that, we prove that the expected number of jobs that are lucky and alive at time t is at least α times the number of jobs that are active at this time (Lemma 5.9), i.e.,

$$\mathbf{E}[\delta^l(t)\mid\mathcal{F}(t,S,\mathbf{x}_{\bar S})]\geq\alpha|S|.$$

For the sake of clarity, the proofs of Lemma 5.8 and Lemma 5.9 are presented in §§5.1.1 and 5.1.2, respectively. We can complete the proof of Lemma 5.5 by using a simple Chernoff bound argument.

Proof of Lemma 5.5 For each $j\in S$, we define $Y_j:=(X_j^l\mid\mathcal{F}(t,S,\mathbf{x}_{\bar S}))$. By Lemma 5.8, the Y_j's are independent. Moreover, $\mathbf{E}[\sum_{j\in S}Y_j]=\mathbf{E}[\delta^l(t)\mid\mathcal{F}(t,S,\mathbf{x}_{\bar S})]\geq\alpha|S|$ by Lemma 5.9. Applying Chernoff's bound (see Appendix A, Theorem A.2), we obtain

$$\mathbf{P}\left[\delta^l(t)<\frac{1}{2}\alpha\delta(t)\mid\mathcal{F}(t,S,\mathbf{x}_{\bar S})\right]=\mathbf{P}\left[\sum_{j\in S}Y_j<\frac{1}{2}\alpha|S|\right]\leq\mathbf{P}\left[\sum_{j\in S}Y_j<\frac{1}{2}\mathbf{E}\left[\sum_{j\in S}Y_j\right]\right]\leq \mathrm{e}^{-\alpha|S|/8}.$$

Finally, summing over all possible subsets $S\subseteq J$ with $|S|=s$ and all possible assignments $\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S}$, the lemma follows. □

In the rest of this section, we only consider properties of the schedule produced by MLF. We therefore omit the superscript MLF in the notation below.

5.1.1. Independence of being lucky. We first study some properties of the probability space conditioned on the event $\mathcal{F}(t,S,\mathbf{x}_{\bar S})=((S(t)=S)\cap(\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S})$ more closely and then prove that the random variables $Y_j=(X_j^l\mid\mathcal{F}(t,S,\mathbf{x}_{\bar S}))$, $j\in S$, are independent.

Lemma 5.6. *Assume $S(t)=S$ and $\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S}$. Then, the schedule of MLF up to time t is uniquely determined.*

Proof. Assume otherwise. Then, there exist two different schedules, $\mathcal{S}_1$ and $\mathcal{S}_2$, such that $S^{\mathcal{S}_1}(t)=S^{\mathcal{S}_2}(t)=S$. Let I_1 and I_2 be the corresponding instances. Since the processing times of jobs not in S are fixed, I_1 and I_2 differ in the processing times of some subset of the jobs in S. Let $t'\leq t$ be the first time where $\mathcal{S}_1$ and $\mathcal{S}_2$ differ. MLF changes its scheduling decision if either (i) a new job is released, or (ii) an active job is completed. Since the release dates are the same in I_1 and I_2, a job j was completed at time t' in one schedule, say $\mathcal{S}_1$, but not in the other. Since j must belong to S and $t'\leq t$, this contradicts the hypothesis that $S^{\mathcal{S}_1}(t)=S$. □

Corollary 1. *Assume $S(t)=S$ and $\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S}$. Then, for each $j\in S$, $x_j^{\text{MLF}}(t)$ is a uniquely determined constant.*

Subsequently, given that $S(t)=S$ and $\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S}$, we set $\pi_j:=x_j^{\text{MLF}}(t)$ for all $j\in S$. We state the following important fact.

Fact 6. *Let I be an instance such that $S(t)=S$ and $\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S}$. Then, every instance I', with $\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S}$ and $p_{jI'}\geq p_{jI}$ for each $j\in S$, satisfies $x_{jI'}^{\text{MLF}}(t)=x_{jI}^{\text{MLF}}(t)$ for each $j\in J$.*

In particular, we can generate all instances satisfying $S(t)=S$ and $\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S}$ as follows. Let I_0 be defined as $\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S}$ and $p_{jI_0}:=\pi_j$ for each $j\in S$. Note that I_0 is not contained in $\mathcal{F}(t,S,\mathbf{x}_{\bar S})$, since $S_{I_0}(t)=\varnothing$; but every instance I with $\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S}$ and $p_{jI}>p_{jI_0}$, for each $j\in S$, is contained in $\mathcal{F}(t,S,\mathbf{x}_{\bar S})$.

Lemma 5.7. *Assume $S(t)=S$ and $\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S}$. Moreover, let $\pi_j=x_j^{\text{MLF}}(t)$ for all $j\in S$. Then, the following events are equivalent*:

$$(S(t)=S)\cap(\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S})\equiv\bigcap_{j\in S}(p_j>\pi_j)\cap(\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S}).$$

Proof. Let I be an instance such that $S(t)=S$ and $\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S}$. By Lemma 5.6, the time spent by MLF on $j\in S$ up to time t is $x_j^{\text{MLF}}(t)=\pi_j$. Since j is active at time t, $p_j>x_j^{\text{MLF}}(t)=\pi_j$.

Next, let I be an instance such that $p_{jI}>\pi_j$ for each $j\in S$ and $\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S}$. Let I_0 be defined as $\mathbf{p}_{\bar S}=\mathbf{x}_{\bar S}$ and $p_{jI_0}:=\pi_j$ for each $j\in S$. For each $j\in S$, we have $p_{jI}>\pi_j=p_{jI_0}$. From the discussion above, we conclude that $I\in\mathcal{F}(t,S,\mathbf{x}_{\bar S})$. □

Lemma 5.8. *The variables $Y_j=(X_j^l\mid\mathcal{F}(t,S,\mathbf{x}_{\bar S}))$, $j\in S$, are independent.*

Proof. Let $R \subseteq S$. For each $j \in R$, let $a_j \in \{0, 1\}$ and let L_j denote the set of processing times such that $(p_j \in L_j)$ if and only if $(X_j^l = a_j)$. From Lemma 5.7, we obtain

$$\begin{aligned}\mathbf{P}\left[\bigcap_{j\in R} X_j^l = a_j \,\middle|\, \mathscr{F}(t, S, \mathbf{x}_{\bar S})\right] &= \mathbf{P}\left[\bigcap_{j\in R} p_j \in L_j \,\middle|\, \bigcap_{j\in S}(p_j > \pi_j) \cap (\mathbf{p}_{\bar S} = \mathbf{x}_{\bar S})\right]\\ &= \frac{\mathbf{P}\left[\bigcap_{j\in R}(p_j \in L_j) \cap \bigcap_{j\in S}(p_j > \pi_j) \cap (\mathbf{p}_{\bar S} = \mathbf{x}_{\bar S})\right]}{\mathbf{P}\left[\bigcap_{j\in S}(p_j > \pi_j) \cap (\mathbf{p}_{\bar S} = \mathbf{x}_{\bar S})\right]}\\ &= \frac{\mathbf{P}\left[\bigcap_{j\in R}(p_j \in L'_j) \cap \bigcap_{j\in S\setminus R}(p_j > \pi_j) \cap (\mathbf{p}_{\bar S} = \mathbf{x}_{\bar S})\right]}{\mathbf{P}\left[\bigcap_{j\in S}(p_j > \pi_j) \cap (\mathbf{p}_{\bar S} = \mathbf{x}_{\bar S})\right]},\end{aligned}$$

where L'_j is defined as the intersection of L_j and $(\pi_j, 2^K]$. Using the fact that processing times are perturbed independently, we obtain

$$\begin{aligned}\mathbf{P}\left[\bigcap_{j\in R} X_j^l = a_j \,\middle|\, \mathscr{F}(t, S, \mathbf{x}_{\bar S})\right] &= \frac{\prod_{j\in R}\mathbf{P}\left[p_j \in L'_j\right]\mathbf{P}\left[\bigcap_{j\in S\setminus R}(p_j > \pi_j) \cap (\mathbf{p}_{\bar S} = \mathbf{x}_{\bar S})\right]}{\prod_{j\in R}\mathbf{P}\left[p_j > \pi_j\right]\mathbf{P}\left[\bigcap_{j\in S\setminus R}(p_j > \pi_j) \cap (\mathbf{p}_{\bar S} = \mathbf{x}_{\bar S})\right]}\\ &= \prod_{j\in R}\frac{\mathbf{P}\left[p_j \in L'_j\right]}{\mathbf{P}\left[p_j > \pi_j\right]} = \prod_{j\in R}\mathbf{P}\left[X_j^l = a_j \mid p_j > \pi_j\right]. \end{aligned} \tag{2}$$

The above equality holds for any subset $R \subseteq S$. In particular, for a singleton set $\{j\}$, we obtain

$$\mathbf{P}\left[X_j^l = a_j \mid \mathscr{F}(t, S, \mathbf{x}_{\bar S})\right] = \mathbf{P}\left[X_j^l = a_j \mid p_j > \pi_j\right]. \tag{3}$$

Therefore, combining (2) and (3), we obtain

$$\mathbf{P}\left[\bigcap_{j\in R} X_j^l = a_j \,\middle|\, \mathscr{F}(t, S, \mathbf{x}_{\bar S})\right] = \prod_{j\in R}\mathbf{P}\left[X_j^l = a_j \mid \mathscr{F}(t, S, \mathbf{x}_{\bar S})\right]. \quad \square$$

5.1.2. Expected number of lucky and alive jobs. From Equation (3) in the proof of Lemma 5.8, we learn that if we concentrate on the probability space conditioned on the event $\mathscr{F}(t, S, \mathbf{x}_{\bar S})$, then

$$\mathbf{P}\left[X_j^l = a_j \mid \mathscr{F}(t, S, \mathbf{x}_{\bar S})\right] = \mathbf{P}\left[X_j^l = a_j \mid p_j > \pi_j\right] \quad \text{for each } j \in S.$$

This relation is very useful in proving the following lemma.

Lemma 5.9. *For every* $j \in S$, $\mathbf{P}\left[X_j^l = 1 \mid \mathscr{F}(t, S, \mathbf{x}_{\bar S})\right] \ge \alpha$. *Thus,* $\mathbf{E}[\delta^l(t) \mid \mathscr{F}(t, S, \mathbf{x}_{\bar S})] \ge \alpha|S|$.

Proof. First, let $\check{p}_j \in (2^{i-1}, 2^i]$ for some integer $i > k$. Due to Fact 3, the processing time p_j is chosen randomly from a subrange of $(2^{i-1}, 2^i]$. Hence,

$$\mathbf{P}\left[X_j^l = 1 \mid \mathscr{F}(t, S, \mathbf{x}_{\bar S})\right] = \mathbf{P}\left[p_j \ge (1+\gamma)2^{i-1} \mid p_j > \pi_j\right] \ge \mathbf{P}\left[\varepsilon_j \ge \gamma 2^{i-1} \mid p_j > \pi_j\right],$$

where the second inequality is due to the fact that $\phi_j \ge 2^{i-1}$. In Appendix D, we show that the events $(\varepsilon_j \ge \gamma 2^{i-1})$ and $(p_j > \pi_j)$ are positively correlated. We therefore have

$$\mathbf{P}\left[X_j^l = 1 \mid \mathscr{F}(t, S, \mathbf{x}_{\bar S})\right] \ge \mathbf{P}\left[\varepsilon_j \ge \gamma 2^{i-1}\right] \ge \mathbf{P}\left[\varepsilon_j \ge (1+\gamma)2^{k-1}\right],$$

where the last inequality holds for every i, $k < i \le K$, if we choose $\gamma \le 2^{k-K}$.

Next, let $\check{p}_j \in [1, 2^k]$. Due to Fact 2, the processing time p_j is chosen completely at random from $[1, 2^k]$. Let L_j denote the set of all processing times such that $(X_j^l = 1)$ holds. Then,

$$\mathbf{P}\left[X_j^l = 1 \mid \mathscr{F}(t, S, \mathbf{x}_{\bar S})\right] = \mathbf{P}\left[\varepsilon_j \in L_j \mid \varepsilon_j > \pi_j\right] \ge \mathbf{P}\left[\varepsilon_j \ge (1+\gamma)2^{k-1}\right].$$

To prove that the last inequality holds, we distinguish two cases:

(a) Let $\pi_j < (1+\gamma)2^{k-1}$. Since $\mathbf{P}\left[\varepsilon_j > \pi_j\right] \le 1$,

$$\mathbf{P}\left[\varepsilon_j \in L_j \mid \varepsilon_j > \pi_j\right] \ge \mathbf{P}\left[(\varepsilon_j \in L_j) \cap (\varepsilon_j > \pi_j)\right] \ge \mathbf{P}\left[\varepsilon_j \ge (1+\gamma)2^{k-1}\right].$$

(b) Let $\pi_j \ge (1+\gamma)2^{k-1}$. Then,

$$\mathbf{P}\left[\varepsilon_j \in L_j \mid \varepsilon_j > \pi_j\right] = 1 \ge \mathbf{P}\left[\varepsilon_j \ge (1+\gamma)2^{k-1}\right].$$

Assuming that the smoothing distribution f satisfies (P1), the lemma follows. $\square$

5.2. Adaptive adversary. Recall that the adaptive adversary may change the input instance on basis of the outcome of the random process. This additional power may affect the correlation technique that we used in Lemmas 5.2 and 5.9. However, as discussed in Appendix D, these lemmas also hold for an adaptive adversary. Thus, the upper bound on the smoothed competitive ratio given in Theorem 5.1 also holds against an adaptive adversary.

6. Lower bounds.

6.1. Lower bounds for the partial bit randomization model. In this section, we present lower bounds on the smoothed competitive ratio for any deterministic algorithm against the oblivious adversary and the stronger, adaptive one. We first proceed with the most intuitive lower bound: the one against the adaptive adversary. The next theorem gives an $\Omega(2^{K-k})$ lower bound on the smoothed competitive ratio under the partial bit randomization model, thus showing that MLF achieves up to a constant factor the best possible ratio in this model. The lower bound is based on ideas similar to those used by Motwani et al. [19] for an $\Omega(2^K)$ nonclairvoyant deterministic lower bound. In the lower bound proofs, we assume that the smoothing distribution is well-shaped with $\mu = 2^{k-1} + 1/2$.

Theorem 6.1. *Any deterministic algorithm* ALG *has smoothed competitive ratio* $\Omega(2^{K-k})$ *against an adaptive adversary in the partial bit randomization model.*

Proof. The input sequence for the lower bound is divided into two phases.

Phase 1. At time $t = 0$, the adversary releases $N := \lfloor (2^{K-k} - 2)/3 \rfloor + 1$ jobs. We run ALG on these jobs up to the first time $\hat{t}$ when a job, say j^*, has been processed for $2^K - 2^{k+1}$ time units. Let $x_j^{\text{MLF}}(\hat{t})$ denote the amount of time spent by algorithm ALG on job j up to time $\hat{t}$. We fix the initial processing time of each job j to $\check{p}_j := x_j^{\text{MLF}}(\hat{t}) + 2^{k+1}$. Note that after smoothing the $\check{p}_j$s, we have $x_j^{\text{MLF}}(\hat{t}) + 2^k < p_j < x_j^{\text{MLF}}(\hat{t}) + 3 \cdot 2^k$ for each j. That is, each job has a remaining processing time between 2^k and $3 \cdot 2^k$. Therefore, ALG will not complete any job at time $\hat{t}$, i.e., $\delta^{\text{ALG}}(\hat{t}) = N$.

Consider the optimal algorithm OPT. If OPT does not process j^* until time $\hat{t}$, $2^K - 2^{k+1}$ time units can be allocated on the other jobs. Thus, at least

$$\frac{2^K - 2^{k+1}}{3 \cdot 2^k} \geq \left\lfloor \frac{2^{K-k} - 2}{3} \right\rfloor = N - 1$$

of these jobs are completed by OPT until time $\hat{t}$, i.e., $\delta^{\text{OPT}}(\hat{t}) = 1$.

Phase 2. The adaptive adversary releases a sequence $N+1, N+2, \ldots$ of jobs. The release time of job $j = N+i$ is $r_j := \hat{t}$ for $i = 1$ and $r_j := r_{j-1} + p_{j-1}$ for $i > 1$. Each such job j has initial processing time $\check{p}_j := 1$ and therefore its smoothed processing time satisfies $p_j \leq 2^k$.

OPT will then complete every job released in the second phase before the next one is released. The optimal strategy for ALG is also to process the jobs released in the second phase to completion as soon as they are released, since every job left uncompleted from the first phase has remaining processing time larger than 2^k.

The second phase goes on for a time interval larger than 2^{3K-2k}, which is an upper bound on the contribution to the total flow time of any algorithm in the first phase of the input sequence. Therefore, in terms of total flow time, the second phase dominates the first phase for both ALG and OPT. Since in the second phase ALG has $\Omega(N)$ jobs and OPT has $O(1)$ jobs in the system, we obtain a competitive ratio of $\Omega(N) = \Omega(2^{K-k})$. □

As mentioned before, the adaptive adversary is stronger than the oblivious one, as it may construct the input instance revealed to the algorithm after time t also, on the basis of the execution of the algorithm up to time t. In the next theorem, we prove a slightly weaker bound of $\Omega(2^{K/6-k/2})$ on the smoothed competitive ratio for any deterministic algorithm against an oblivious adversary under the partial bit randomization smoothing model.

Theorem 6.2. *Any deterministic algorithm* ALG *has smoothed competitive ratio* $\Omega(2^{K/6-k/2})$ *for every* $k \leq K/3$ *against an oblivious adversary in the partial bit randomization model.*

Proof. For notational convenience, we assume that K is even. The input sequence for the lower bound is divided into two phases.

Phase 1. At time $t = 0$, the adversary releases $N := 2^{K/2} + \lfloor (2^{K-k} - 2)/3 \rfloor$ jobs and runs ALG on these jobs up to the first time $\hat{t}$ when one of the following two events occurs: (i) $2^{K/2}$ jobs, denoted by $j_1^*, j_2^*, \ldots, j_{2^{K/2}}^*$, have been processed for at least $2^{K/2}$ time units, or (ii) one job, say j^*, has been processed for $2^K - 2^{k+1}$ time units. In the sequel, we call jobs released in the first phase *phase*-1 *jobs*.

Let $x_j^{\text{MLF}}(\hat{t})$ denote the amount of time spent by algorithm ALG on job j up to time $\hat{t}$. We fix the initial processing time of each job j to $\check{p}_j := x_j^{\text{MLF}}(\hat{t}) + 2^{k+1}$. Note that after smoothing the $\check{p}_j$s, we have $x_j^{\text{MLF}}(\hat{t}) + 2^k < p_j < x_j^{\text{MLF}}(\hat{t}) + 3 \cdot 2^k$ for each j. That is, in the schedule produced by ALG, each job has a remaining processing time between 2^k and $3 \cdot 2^k$ at time $\hat{t}$. Moreover, ALG has not completed any job at this time, i.e., $\delta^{\text{ALG}}(\hat{t}) = N$.

Instead of considering an optimal scheduling algorithm, we consider a scheduling algorithm $\mathcal{S}$ that schedules the jobs as described below. Clearly, the total flow time of OPT is upper bounded by the total flow time of $\mathcal{S}$.

Let $\hat{t}$ be determined by case (i). Then, $\mathcal{S}$ does not process jobs $j_1^*, j_2^*, \ldots, j_{2^{K/2}}^*$ before all other jobs are completed. Therefore, at least 2^K time units can be allocated on the other jobs. Since each of these $N - 2^{K/2}$ jobs has remaining processing time at most $3 \cdot 2^k$, $\mathcal{S}$ has completed at least

$$\min\left(N - 2^{K/2}, \left\lfloor \frac{2^K}{3 \cdot 2^k} \right\rfloor\right) \geq N - 2^{K/2}$$

jobs, i.e., all these jobs. In case (ii), by not processing job j^*, $\mathcal{S}$ completes at least

$$\min\left(N - 1, \left\lfloor \frac{2^K - 2^{k+1}}{3 \cdot 2^k} \right\rfloor\right) \geq N - 2^{K/2}$$

of the other jobs. Thus, we obtain $\delta^{\mathcal{S}}(\hat{t}) \leq 2^{K/2}$.

Phase 2. Starting from time $\hat{t}$, the adversary releases a sequence of $L := 2^{5K/3-k}$ jobs, denoted by $N+1, N+2, \ldots, N+L$, for a period of $\tilde{t} := \mu L$, where $\mu := 2^{k-1} + \frac{1}{2}$. The release time of job $j = N+i$ is $r_j := \hat{t} + (i-1)\mu$, for $i = 1, \ldots, L$. Each such job j has initial processing time $\check{p}_j := 1$ and its smoothed processing time satisfies $p_j \leq 2^k$. In the sequel, we call jobs released in the second phase *phase-2 jobs*.

To analyze the number of jobs in the system of ALG and $\mathcal{S}$ during the second phase, we define the random variables $X_j := p_{N+j} - \mu$, for $j = 1, \ldots, L$. Note that the X_js are independently distributed random variables with zero mean. Define $S_0 := 0$ and $S_i := \sum_{j=1}^{i} X_j$, for $i = 1, \ldots, L$. Applying Kolmogorov's inequality (see Appendix A, Theorem A.1), we obtain

$$\mathbf{P}\left[\max_{0 \leq i \leq L} |S_i| \geq \mu\sqrt{L}\right] \leq \frac{\mathbf{E}[S_L^2]}{\mu^2 L} \leq \frac{1}{3} \tag{4}$$

The last inequality follows since $\mathbf{E}[S_L^2] = \mathbf{Var}[S_L]$ and the variance of the random variable S_L for the uniform distribution is $L(2^{2k} - 1)/12$. The bound holds for any well-shaped distribution, since among these distributions the variance is maximized by the uniform distribution.

Consider a schedule $\mathcal{Q}$ only processing phase-2 jobs. The amount of idle time up to time $\hat{t} + i\mu$ is given by

$$I_0 := 0 \quad \text{and} \quad I_i := \max\left(I_{i-1}, i\mu - \sum_{j=1}^{i} p_{N+j}\right).$$

Hence, the total idle time up to time $\hat{t} + i\mu$ for this algorithm is

$$I_i = \max_{0 \leq j \leq i} -S_j.$$

By (4), we know that with probability at least $\frac{2}{3}$ the total idle time at any time $\hat{t} + i\mu$ stays below $\mu\sqrt{L}$.

We first derive a lower bound on the number of jobs that are in the system of ALG during the second phase.

LEMMA 6.1. *With probability at least $\frac{2}{3}$, at any time $t \in [\hat{t}, \hat{t} + \tilde{t}]$: $\delta^{\text{ALG}}(t) \geq N - \frac{1}{2}\sqrt{L} - 1$.*

PROOF. ALG can do no better than the SRPT rule during the second phase. Each phase-1 job has remaining processing time larger than 2^k. Therefore, ALG follows $\mathcal{Q}$ using the idle time to schedule phase-1 jobs, unless a phase-1 job has received so much processing time that its remaining processing time is less than the processing time of the newly released job. This leads to at most an additional 2^k time spent on phase-1 jobs. Hence, with probability at least $\frac{2}{3}$, at most $\frac{1}{2}\sqrt{L} + 1$ phase-1 jobs are finished by ALG during the second phase. □

$\mathcal{S}$ also follows $\mathcal{Q}$ during the second phase using the idle time to schedule phase-1 jobs. We next give an upper bound on the number of jobs in the system of $\mathcal{S}$ during the second phase.

LEMMA 6.2. *With probability at least $\frac{2}{3}$, at any time $t \in [\hat{t}, \hat{t} + \tilde{t}]: \delta^{\mathcal{S}}(t) \leq 2^{K/2} + 2\sqrt{L} + 2$.*

Proof. Consider the amount of additional volume brought into the system. Just before time $t = \hat{t} + i\mu$, this is

$$\sum_{j=1}^{i} p_j - (i\mu - I_i),$$

i.e., the total processing time of phase-2 jobs released before time t minus the amount of time processed on phase-2 jobs. Hence, the maximum amount of additional volume before the release of a phase-2 job is given by

$$\Delta V = \max_{0\le i\le L}(S_i + I_i) = \max_{0\le i\le L}(S_i + \max_{0\le j\le i} -S_j) = \max_{0\le j\le i\le L}(S_i - S_j).$$

The probability that this value exceeds some threshold value is bounded by

$$\mathbf{P}[\Delta V > 2\lambda] \le \mathbf{P}\left[\max_{0\le i,\, j\le L}(S_i - S_j) > 2\lambda\right] \le \mathbf{P}\left[\max_{0\le i\le L}|S_i| > \lambda\right].$$

Setting λ to $\mu\sqrt{L}$, by (4) this probability is at most $\frac{1}{3}$.

To conclude the proof we need the following fact, which can easily be proven by induction on the number of phase-2 jobs released.

Fact 7. *Just before the release of a phase-*2 *job, $\mathscr{S}$ has no more than one phase-*2 *job with remaining processing time less than μ.*

Assume ΔV attains its maximum just before time $t' := \hat{t} + i\mu$. Due to Fact 7, no more than one phase-2 job has remaining processing time less than μ. At time t', a new phase-2 job is released. Therefore, with probability at least $\frac{2}{3}$, the number of phase-2 jobs that are in the system is bounded by

$$\frac{2\mu\sqrt{L}}{\mu} + 2 = 2\sqrt{L} + 2. \quad \square$$

By the above two lemmas, with constant probability the total flow time of the two schedules is bounded by

$$F^{\text{ALG}} \ge \left(N - \sqrt{L}/2 - 1\right)\tilde{t},$$
$$F^{\mathscr{S}} \le N\hat{t} + \left(2^{K/2} + 2\sqrt{L} + 2\right)\tilde{t} + \left(2^{K/2} + 2\sqrt{L} + 2\right)\left(3N2^k + 2\mu\sqrt{L}\right),$$

where the contribution of the period after time $\hat{t} + \tilde{t}$ for $\mathscr{S}$ is bounded by the number of jobs at time $\hat{t} + \tilde{t}$ times the remaining processing time at the start of this phase.

To bound the ratio between F^{ALG} and $F^{\mathscr{S}}$, we note that by definition of N and as $\hat{t} \le 2^{K/2}(2^K - 2^{k+1}) + (N - 2^{K/2})2^{K/2}$, it follows that $N\hat{t} \le 2(2^{K/2} + 2\sqrt{L} + 2)\mu L$. Moreover, we know from the definition of N and μ that $3N2^k + 2\mu\sqrt{L} \le 8\mu L$. Hence, by restricting $k \le K/3$, we have that

$$\mathbf{E}\left[\frac{F^{\text{ALG}}}{F^{\text{OPT}}}\right] = \Omega\left(\frac{N - \sqrt{L}/2 - 1}{2^{K/2} + 2\sqrt{L} + 2}\right) = \Omega\left(\frac{2^{K-k} + 2^{K/2} - 2^{5K/6-k/2}}{2^{5K/6-k/2}}\right) = \Omega\left(2^{K/6-k/2}\right). \quad \square$$

6.2. Lower bounds for symmetric smoothing models. Since we are using the partial bit randomization model, we do not smoothen the processing times symmetrically around their initial values. Therefore, a natural question is whether or not symmetric smoothing models, as defined in §3, are more suitable to analyze MLF. We answer this question in the negative by providing a lower bound of $\Omega(2^K)$ on the performance of MLF under the following symmetric smoothing model.

Consider a function $\vartheta : \mathbb{R}^+ \to \mathbb{R}^+$ which is continuous and nondecreasing. In the symmetric smoothing model, according to ϑ we smoothen the initial processing times as follows:

$$p_j := \max(1, \check{p}_j + \varepsilon_j), \quad \text{where } \varepsilon_j \overset{f}{\leftarrow} [-\vartheta(\check{p}_j)/2, \vartheta(\check{p}_j)/2],$$

and f is the uniform distribution.

In the following theorem, we prove that for certain functions ϑ, the smoothed competitive ratio against an oblivious adversary can be as bad as $\Omega(2^K)$. As for the previous two lower bounds, we define a two-phase sequence. The jobs released in the first phase should be large enough such that they all enter Q_{K-1}. Moreover, with constant probability, a large enough fraction of these jobs should enter queue Q_K.

Theorem 6.3. *Let $\vartheta : \mathbb{R}^+ \to \mathbb{R}^+$ be a continuous and nondecreasing function such that there exists a $x^* \in \mathbb{R}^+$ satisfying $x^* - \vartheta(x^*)/2 > 2^{K-2}$ and $x^* + \vartheta(x^*)/2 = 2^{K-1} + a$ for some constant $1 \le a \le 2^{K-1}$. Then, there exists an $\Omega(2^K/a)$ lower bound on the smoothed competitive ratio of MLF against an oblivious adversary in the symmetric smoothing model according to ϑ.*

The symmetric smoothing model according to ϑ captures the additive symmetric, the additive relative symmetric, and the relative smoothing model, which can be seen as follows.

The additive symmetric smoothing model over $[-c, c]$ is equivalent to the above defined model with $\vartheta(x) := 2c$. Since $x^* - c > 2^{K-2}$ and $x^* = 2^{K-1} + a - c$, we obtain $c < 2^{K-3} + a/2$. By fixing $a := 1$, Theorem 6.3 yields an $\Omega(2^K)$ lower bound for the symmetric additive smoothing model against an oblivious adversary.

For the additive relative symmetric smoothing model, we define $\vartheta(x) := 2x^c$, for $c \geq 0$. From the condition $x^* - (x^*)^c > 2^{K-2}$ and the definition of x^*, we obtain $(x^*)^c < 2^{K-3} + a/2$. We fix $a := 1$ and require $(x^*)^c \leq 2^{K-3}$, and thus $c \leq (K-3)/\log(2^{K-1}+1)$. Theorem 6.3 then yields an $\Omega(2^K)$ lower bound for the additive relative symmetric smoothing model.

The relative smoothing model is equivalent to the symmetric smoothing model according to ϑ with $\vartheta(x) := 2\epsilon x$. The conditions in Theorem 6.3 are fulfilled if $0 \leq \epsilon \leq (2^{K-2} + a)/(3 \cdot 2^{K-2} + a)$. Hence, for $a := 1$, we obtain an $\Omega(2^K)$ lower bound for the relative smoothing model.

Proof of Theorem 6.3. The input sequence of the adversary consists of two phases. Let $\mathcal{S}$ be the algorithm that during the first phase schedules the jobs to completion in the order in which they are released, and during the second phase schedules the jobs that are released in this phase to completion in the order in which they are released. After having completed all phase-2 jobs, $\mathcal{S}$ finishes the remaining phase-1 jobs. We upper bound OPT by $\mathcal{S}$. To prove the theorem, we show that with constant probability $F^{\text{MLF}}/F^{\mathcal{S}} = \Omega(2^K/a)$. Then $\mathbf{E}[F^{\text{MLF}}/F^{\text{OPT}}] = \Omega(2^K/a)$. Without loss of generality, we assume that $K \geq 3$, and we define $L := \vartheta(x^*)$.

Phase 1. At time $t = 0$, $M := 8\max(L^3/2^K, 1)$ jobs are released with initial processing time $\check{p}_1 := x^*$, and then every $\check{p}_1$ time units one job with the same initial processing time is released. The total number of jobs released in the first phase is $N := \max(L^4, 2^{2K}/L^2)$. Note that by definition of x^*, the smoothed processing time of each phase-1 job is at least 2^{K-2}.

Let $T_1(i)$ be the total processing time of jobs released in phase 1 at or before time $i\check{p}_1$, for $i = 0, 1, \ldots, N-M$. Define $S_0 := 0$ and $S_i := S_{i-1} + \varepsilon_i = \sum_{j=1}^{i} \varepsilon_j$, for $i = 1, \ldots, N$. As $\mathbf{E}[\varepsilon_j] = 0$ and all ε_j are drawn independently, we have $\mathbf{E}[S_i] = 0$ and $\mathbf{E}[S_i^2] = iL^2/12$, for all $i = 0, \ldots, N$. Applying Kolmogorov's inequality (see Appendix A, Theorem A.1), we obtain

$$\left[\max_{0\leq k\leq N} |S_k| > L\sqrt{N}\right] \leq \tfrac{1}{12}.$$

Hence, we have with probability at least 11/12 that for all $i = 0, \ldots, N-M$

$$(i+M)\check{p}_1 - L\sqrt{N} \leq T_1(i) \leq (i+M)\check{p}_1 + L\sqrt{N}. \tag{5}$$

In the sequel, we assume that (5) holds.

Let $\hat{t} := (N-M+1)\check{p}_1$, and consider a $t \in [0, \hat{t})$. Then, the remaining processing time for $\mathcal{S}$ as well as MLF at time t is

$$\begin{aligned} T_1(\lfloor t/\check{p}_1\rfloor) - t &\geq (\lfloor t/\check{p}_1\rfloor + M)\check{p}_1 - L\sqrt{N} - t \\ &\geq t - \check{p}_1 + M\check{p}_1 - L\sqrt{N} - t \geq (M-1)2^{K-2} - L\sqrt{N} - 1 \\ &\geq 2\max(L^3, 2^K) - \max(L^3, 2^K) - 2^{K-2} - 1 > 0. \end{aligned} \tag{6}$$

Hence, $\mathcal{S}$ and MLF do not have any idle time during the first phase. Moreover, the remaining processing time for both algorithms is at most $M\check{p}_1 + L\sqrt{N}$.

Consider some $t \in [0, \hat{t})$. There is at most one job that has been processed on by $\mathcal{S}$ but is not yet completed. Hence,

$$\delta^{\mathcal{S}}(t) \leq \frac{M\check{p}_1 + L\sqrt{N}}{2^{K-2}} + 1 = O(M).$$

Consider the schedule produced by MLF up to time $\hat{t}$. The probability that a job released in phase 1 is of class K is at least a/L. The expected number of phase-1 class K jobs is at least aN/L. Applying Chernoff's bound (Appendix A, Theorem A.2), we know that with probability at least $1 - \mathrm{e}^{-aN/(8L)} \geq (\mathrm{e}-1)/\mathrm{e}$, there are at least $aN/(2L)$ class K phase-1 jobs. In the sequel, we assume that this property holds. Note that the probability that both (5) and the bound on the number of class K jobs hold is at least $(\mathrm{e}-1)/\mathrm{e} - 1/12$.

If MLF does not finish any class K job up to time $\hat{t}$, then

$$\delta^{\text{MLF}}(\hat{t}) \geq \frac{aN}{2L}.$$

Otherwise, consider the last time $t \in [0, \hat{t})$ that MLF was processing a job in queue Q_K. By definition of MLF, we know that at this time all lower queues were empty. Moreover, we know that the remaining processing time of each job in this queue is at most a, and we also know from (6) that the total remaining processing time is at least $\max(L^3, 2^K) - 1 = L\sqrt{N} - 1$. Hence, at this time, the number of alive jobs in the schedule of MLF is at least $(L\sqrt{N}-1)/a$, and also

$$\delta^{\text{MLF}}(\hat{t}) \geq \frac{L\sqrt{N}-1}{a}.$$

Phase 2: At time $\hat{t}$, M jobs with $\check{p}_2 := 2^{K-2}$ are released and then, every $\check{p}_2$ time units, one job with the same $\check{p}_2$ is released. The total number of jobs released in this phase is $2N$. As $\vartheta(\check{p}_2)/2 \leq a/2 \leq 2^{K-2}$, no job released in the second phase enters queue Q_K.

Let $T_2(i)$ be the total processing time of the phase-2 jobs released at or before time $\hat{t} + i\check{p}_2$. Applying Kolmogorov's inequality yields that with probability at least 11/12, we have

$$(i+M)\check{p}_2 - L\sqrt{2N} \leq T_2(i) \leq (i+M)\check{p}_2 + L\sqrt{2N}. \tag{7}$$

In the sequel, we assume that also (7) holds. The probability that the bound on the number of class K jobs and (5) and (7) hold is at least $(\mathrm{e}-1)/\mathrm{e} - 1/6 > 0.46$.

Using the same arguments as before, we now show that MLF continuously processes phase-2 jobs until time $\bar{t} := \hat{t} + (2N - M + 1)\check{p}_2$. Namely, consider a $t \in [\hat{t}, \bar{t})$. Then, the remaining processing time for $\mathcal{S}$ as well as MLF at time t is

$$\begin{aligned} T_2(\lfloor (t-\hat{t})/\check{p}_2 \rfloor) - (t - \hat{t}) &\geq (\lfloor (t-\hat{t})/\check{p}_2 \rfloor + M)\check{p}_2 - L\sqrt{2N} - (t-\hat{t}) \\ &\geq (M-1)\check{p}_2 - L\sqrt{2N} - 1 \geq (M-1)2^{K-2} - L\sqrt{2N} - 1 \\ &\geq 2\max(L^3, 2^K) - \sqrt{2}\max(L^3, 2^K) - 2^{K-2} - 1 > 0. \end{aligned}$$

Thus, if MLF does not finish any phase-1 job of class K up to time $\hat{t}$, we have

$$\delta^{\text{MLF}}(t) \geq \frac{aN}{2L}, \quad \text{for } t \in [\hat{t}, \bar{t}), \quad \text{and} \quad F^{\text{MLF}} = \Omega\left(\frac{aN}{2L}(2N - M + 1)\check{p}_2\right).$$

Otherwise, we have

$$\delta^{\text{MLF}}(t) \geq \frac{L\sqrt{N}-1}{a}, \quad \text{for } t \in [\hat{t}, \bar{t}), \quad \text{and} \quad F^{\text{MLF}} = \Omega\left(\frac{L\sqrt{N}}{a}(2N - M + 1)\check{p}_2\right).$$

Moreover, using the same argumentation as for phase 1, we know that during $[\hat{t}, \bar{t})$, $\mathcal{S}$ has at most $(M\check{p}_2 + L\sqrt{2N})/2^{K-3} + 1 = (2 + \sqrt{2})M + 1$ phase-2 jobs in its system. Hence,

$$\delta^{\mathcal{S}}(t) = O(M) \quad \text{for } t \in [\hat{t}, \bar{t}).$$

After time $\bar{t}$, the time needed by $\mathcal{S}$ to finish all jobs is at most

$$M\check{p}_1 + L\sqrt{N} + M\check{p}_2 + L\sqrt{2N} \leq \left(\frac{9+\sqrt{2}}{2} + 1\right)M\check{p}_2 \leq \left(\frac{9+\sqrt{2}}{2} + 1\right)(2N - M + 1)\check{p}_2.$$

Hence,

$$F^{\mathcal{S}} = O(M(2N - M + 1)\check{p}_2).$$

If $N = L^4$, then $M = 8L^3/2^K$ and

$$F^{\text{MLF}}/F^{\mathcal{S}} = \Omega\left(\frac{aN}{2LM}\right) = \Omega(a2^K) \quad \text{or} \quad F^{\text{MLF}}/F^{\mathcal{S}} = \Omega\left(\frac{L\sqrt{N}}{M}\right) = \Omega\left(\frac{2^K}{a}\right).$$

If $N = 2^{2K}/L^2$, then $L^3 \leq 2^K$ and $M = 8$. Moreover,

$$F^{\text{MLF}}/F^{\mathcal{S}} = \Omega\left(\frac{aN}{2LM}\right) = \Omega(a2^K) \quad \text{or} \quad F^{\text{MLF}}/F^{\mathcal{S}} = \Omega\left(\frac{L\sqrt{N}}{M}\right) = \Omega\left(\frac{2^K}{a}\right).$$

Since the probability that (5), (7), and the bound on the number of class K jobs hold is constant and $a \geq 1$, we have

$$\mathbf{E}\left[\frac{F^{\text{MLF}}}{F^{\text{OPT}}}\right] = \Omega\left(\frac{2^K}{a}\right). \quad \square$$

Obviously, Theorem 6.3 also holds for the adaptive adversary. Finally, we remark that we can generalize the theorem to the case that f is a well-shaped function.

6.3. Lower bound when smoothing release dates and processing times. We present a lower bound of $\Omega(2^{K-k})$ on the smoothed competitive ratio of MLF under the partial bit randomization model, if the release dates are perturbed additionally. The lower bound holds for any smoothing model of release dates that satisfies $|r_j - \check{r}_j| \le 2^{K-1}$. We may even allow negative release dates.

Theorem 6.4. *Let the processing times be smoothed according to the partial bit randomization model, and the release dates be smoothed such that the disruption is no more than 2^{K-1}. Then, there exists an $\Omega(2^{K-k})$ lower bound on the smoothed competitive ratio of MLF.*

Proof. The instance consists of $n+3$ jobs, where $n := 2^{K-k}$. All jobs have original processing time $2^{K-1}+1$, i.e., the smoothed processing time p_j is in $[2^{K-1}+1, 2^{K-1}+2^k]$. The first three jobs, denoted by $-2, -1, 0$, are originally released at time $\check{r}_j = 0$. The other jobs have original release date $\check{r}_j := j \cdot 2^{K-1}$, $j = 1, \dots, n$.

Lemma 6.3. *Let $r_{\max} := \max_{-2\le j\le n} r_j$ be the largest smoothed release date. MLF does not finish any of the jobs $j \ge 1$ until time $r_{\max} + 2^{K-1}$.*

Proof. Any job will be completed in queue Q_K, and thus we have to prove that when a job $j \ge 1$ has been processed for in total 2^{K-1} time units, another job has been released. Therefore, no job $j \ge 2$ will be processed in Q_K until all jobs have been released.

Let r_1 be the smoothed release date of job 1. If $0 \le r_1 < \max(r_{-2}, r_{-1}, r_0) \le 2^{K-1}$, then a job is released within 2^{K-1} time units after the release of job $j = 1$. Suppose that $r_2 \ge r_1 \ge \max(r_{-2}, r_{-1}, r_0)$. Then, job 1 enters Q_K, not before $\min(r_{-2}, r_{-1}, r_0) + 4 \cdot 2^{K-1} \ge 3 \cdot 2^{K-1}$ (due to possible negative release dates of the first three jobs). As the smoothed release date of job $j = 2$ is $r_1 \le r_2 \le 3 \cdot 2^{K-1}$, job $j = 2$ is released before MLF can start processing job 1 in queue Q_K. Suppose that $r_2 < r_1$, then job $j = 1$ has been processed for 2^{K-1} time units not before $\min(r_{-2}, r_{-1}, r_0) + 5 \cdot 2^{K-1} \ge 4 \cdot 2^{K-1}$, and this is the latest possible release date for job $j = 3$.

We can repeat this argument for all jobs $j > 1$. □

From Lemma 6.3, it follows that in the interval $[j2^{K-1}, (j+1)2^{K-1})$, for $j = 1, \dots, n-1$, at least j jobs are alive in the schedule of MLF. Hence, the total flow time for MLF is bounded from below by

$$F^{\text{MLF}} \ge \tfrac{1}{2} n(n-1)2^{K-1}.$$

Lemma 6.4. *If $n \le 2^{K-k}$, then SRPT has at each time a constant number of jobs.*

Proof. Up to time 2^{K-1}, no more than four jobs have been released. Consider a time $i \cdot 2^{K-1} \le t < (i+1)2^{K-1}$. No more than $(i+4)$ jobs have been released and each has processing time at least $2^{K-1}+1$ and at most $2^{K-1}+2^k$. That is, at time t, the total remaining processing time is at most $(i+4)(2^{K-1}+2^k) - (t - 2^{K-1}) \le 5 \cdot 2^{K-1} + (i+4)2^k \le 11 \cdot 2^{K-1}$, for $i \le 2^{K-k}$ and $k \le K-1$. Hence, in the optimal schedule, no more than 11 jobs are alive. □

From this lemma, it follows that, if $k \le K-1$, the total flow time in the optimal schedule is no more than

$$F^{\text{OPT}} \le 11(2^{K-k}+3)(2^{K-1}+2^k) \le 11 \cdot 2^{K-k}2^{K-1} + 88 \cdot 2^{K-1} \le 99 \cdot 2^{K-k}2^{K-1}.$$

Hence, the ratio between F^{MLF} and F^{OPT} is bounded from below by

$$\frac{F^{\text{MLF}}}{F^{\text{OPT}}} \ge \frac{0.5 \cdot 2^{K-k}(2^{K-k}-1)2^{K-1}}{99 \cdot 2^{K-k}2^{K-1}} \ge c \cdot 2^{K-k},$$

for some constant c. □

7. Concluding remarks.

We analyzed the performance of the multilevel feedback algorithm using the novel approach of smoothed analysis. Smoothed competitive analysis provides a unifying framework for worst-case and average-case analysis of online algorithms. We considered several smoothing models, including the additive symmetric smoothing model proposed by Spielman and Teng [25]. The partial bit randomization model yields the best upper bound.

In particular, we proved that the smoothed competitive ratio of MLF using this model is $O((2^k/\sigma)^3 + (2^k/\sigma)^2 2^{K-k})$, where σ is the standard deviation of the smoothing distribution. The analysis holds for various distributions. For distributions with $\sigma = \Theta(2^k)$, e.g., for the uniform distribution, we obtain a smoothed competitive ratio of $O(2^{K-k})$. By choosing $k = K$, the result implies a constant upper bound on the average competitive ratio of MLF. We also proved that any deterministic algorithm has smoothed competitive ratio $\Omega(2^{K-k})$. Hence, in this model, MLF is optimal up to a constant factor. Moreover, we showed that $\Omega(2^{K-k})$ is a lower bound for the smoothed competitive ratio of MLF if the release dates are smoothed additionally. For the other proposed smoothing models, we have obtained lower bounds of $\Omega(2^K)$. Thus, these models do not seem to capture the good performance of MLF in practice.

As mentioned in the introduction, one could alternatively define the smoothed competitive ratio as the ratio between the expected cost of the algorithm and the expected optimal cost (Scharbrodt et al. [21]), rather than the expected competitive ratio. We remark that from Lemmas 5.1, 5.2, and 5.9, we obtain the same bound under this alternative definition, without the need for any high-probability argument.

Interesting open problems are the analysis of MLF when the release times of the jobs are smoothed, and to improve the lower bound against the oblivious adversary in the partial bit randomization model. It can also be of some interest to extend our analysis to the multiple machine case. Following the work of Becchetti and Leonardi [3], we can extend Lemma 5.1 having an extra factor of K, which will also be in the smoothed competitive ratio. Finally, we hope that this framework of analysis will be extended to other online problems.

Appendix A. Bounds on large deviations. For the sake of completeness, we state several well-known results that we will use in the paper. The first is known as Kolmogorov's inequality; see, e.g., Feller [11].

Theorem A.1. *Let $X_1, \ldots, X_n$ be a sequence of independent random variables such that $\mathbf{E}[X_j] = 0$ for all j. Define $S_0 := 0$ and $S_i := \sum_{j\le i} X_j$. Then,*

$$\mathbf{P}\left[\max_{0\le k\le n} |S_k| \ge \lambda\right] \le \frac{\mathbf{E}[S_n^2]}{\lambda^2} \quad \textit{for any } \lambda > 0.$$

We will also use the following versions of Chernoff bounds.

Theorem A.2. *Let X be the sum of a finite number of mutually independent binary random variables such that $\mu := \mathbf{E}[X]$ is positive. Then,*

$$\mathbf{P}[X \le (1-\delta)\mu] < \mathrm{e}^{-\mu\delta^2/2} \quad \textit{for any } \delta \in \mathbb{R}^+ \textit{ with } \delta < 1.$$

Theorem A.3. *Let X be the sum of a finite number of mutually independent binary random variables such that $\mu := \mathbf{E}[X]$ is positive. Then,*

$$\mathbf{P}[X \ge (1+\delta)\mu] < \left(\frac{\mathrm{e}^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu} \quad \textit{for any } \delta \in \mathbb{R}^+.$$

Theorem A.4. *Let X be the sum of a finite number of mutually independent binary random variables such that $\mu := \mathbf{E}[X]$ is positive. Then,*

$$\mathbf{P}[|X-\mu| > \delta\mu] < 2\mathrm{e}^{-\mu\delta^2/3} \quad \textit{for any } \delta \in \mathbb{R}^+.$$

We also use the following bound, known as the Hoeffding bound.

Theorem A.5. *Let $X_1, \ldots, X_n$ be independent random variables. Define $X := \sum_{i\in[n]} X_i$ and $\mu := \mathbf{E}[X]$. If each $X_i \in [a_i, b_i]$, $i \in [n]$, for some constants a_i and b_i, then, for any $t > 0$,*

$$\mathbf{P}[X \le \mathbf{E}[X] - t] \le \exp\left(\frac{-2t^2}{\sum_i (b_i - a_i)^2}\right), \quad \textit{and}$$

$$\mathbf{P}[X \ge \mathbf{E}[X] + t] \le \exp\left(\frac{-2t^2}{\sum_i (b_i - a_i)^2}\right).$$

Appendix B. Characterization of feasible smoothing distributions. In the following, we attempt to characterize distributions that satisfy properties (P1)–(P3).

We start with (P1). A trivial lower bound on the tail probability $\mathbf{P}\left[\varepsilon \ge (1+\gamma)2^{k-1}\right]$ is given by the following lemma, where we assume a uniform distribution over $[1, (1+\gamma)2^{k-1})$. We remark that although Lemma B.1 is straightforward, it might be indeed tight, e.g., for the uniform distribution.

Lemma B.1. *Let ε be a random variable chosen according to a distribution f over $[1, 2^k]$. Moreover, let M be such that $\mathbf{P}[\varepsilon = x] \le M$ for each $x \in [1, (1+\gamma)2^{k-1})$. Then, $\mathbf{P}\left[\varepsilon \ge (1+\gamma)2^{k-1}\right] \ge 1 - M(1+\gamma)2^{k-1}$.*

Proof.

$$\mathbf{P}\left[\varepsilon \ge (1+\gamma)2^{k-1}\right] = 1 - \mathbf{P}\left[\varepsilon < (1+\gamma)2^{k-1}\right] \ge 1 - M(1+\gamma)2^{k-1}. \quad \square$$

We also obtain two other lower bounds on the tail probability of f. Both use an "inverse" version of Chebyshev's inequality. We first prove the following lemma; see also Grimmett and Stirzaker [12].

Lemma B.2. *Let ε be a random variable and let $h(\varepsilon)$ be a nonnegative function such that $h(\varepsilon) \le M$ for each ε. Then,*

$$\mathbf{P}[h(\varepsilon) > \lambda] \ge \frac{\mathbf{E}[h(\varepsilon)] - \lambda}{M - \lambda}.$$

Proof. Let $X_{h(\varepsilon)}$ be 1 if $(h(\varepsilon) > \lambda)$ and 0 otherwise. We have

$$h(\varepsilon) \le M \cdot X_{h(\varepsilon)} + \lambda \cdot (1 - X_{h(\varepsilon)}),$$

and by linearity of expectation,

$$\mathbf{E}[h(\varepsilon)] \le M \cdot \mathbf{E}[X_{h(\varepsilon)}] + \lambda \cdot (1 - \mathbf{E}[X_{h(\varepsilon)}]).$$

The proof now follows from the fact that $\mathbf{E}[X_{h(\varepsilon)}] = \mathbf{P}[h(\varepsilon) > \lambda]$. □

We are now in a position to obtain our first inverse Chebyshev inequality.

Lemma B.3 (Inverse Chebyshev Inequality I). *Let ε be a random variable chosen according to a distribution f over $[1, 2^k]$ with mean μ and standard deviation σ. Then, for each $0 < \lambda < 2^k$,*

$$\mathbf{P}[\varepsilon > \lambda] \ge \frac{\sigma^2 + \mu^2 - \lambda^2}{2^{2k} - \lambda^2}.$$

Proof. Define $h(\varepsilon) := \varepsilon^2$. Then, $h(\varepsilon) \le 2^{2k}$ for each ε. The bound now follows from Lemma B.2, where we exploit that $\sigma^2 = \mathbf{E}[\varepsilon^2] - \mu^2$. □

The following lemma shows that for $\gamma := 2^{k-K-1}$, we obtain $\alpha = (\sigma/2^k)^2$, if only the expectation of f is large enough. We remark that the requirement on δ is always satisfied if $\mu \ge \frac{3}{4} \cdot 2^k$.

Lemma B.4. *Let ε be a random variable chosen according to a distribution f over $[1, 2^k]$ with mean $\mu \ge \delta \cdot 2^k$ and standard deviation σ. Define $\gamma := 2^{k-K-1}$. If $\delta \ge \frac{1}{2}(1+\gamma)$, then $\mathbf{P}[\varepsilon \ge (1+\gamma)2^{k-1}] \ge (\sigma/2^k)^2$.*

Proof. The proof follows from Lemma B.3 and since $\mu \ge \delta \cdot 2^k \ge (1+\gamma)2^{k-1}$. □

We derive our second inverse Chebyshev inequality.

Lemma B.5 (Inverse Chebyshev Inequality II). *Let ε be a random variable chosen according to a distribution f over $[1, 2^k]$ with mean μ and standard deviation σ. Then, for each $0 < \lambda < 2^k - \mu$,*

$$\mathbf{P}[|\varepsilon - \mu| \ge \lambda] \ge \frac{\sigma^2 - \lambda^2}{(2^k - \mu)^2 - \lambda^2}.$$

Proof. Define $h(\varepsilon) := (\varepsilon - \mu)^2$. Then $h(\varepsilon) \le (2^k - \mu)^2$ for each ε. The proof follows from Lemma B.2. □

The next lemma applies if the underlying distribution f satisfies $\mathbf{P}[\varepsilon \ge \mu + \sigma/\sqrt{2}] \ge \mathbf{P}[\varepsilon \le \mu - \sigma/\sqrt{2}]$. For example, this condition holds if f is symmetric around μ or if f is nondecreasing over $[1, 2^k]$.

Lemma B.6. *Let ε be a random variable chosen according to a distribution f over $[1, 2^k]$ with mean $\mu \ge \delta \cdot 2^k$ and standard deviation σ, and assume $\mathbf{P}[\varepsilon \ge \mu + (\sigma/\sqrt{2})] \ge \mathbf{P}[\varepsilon \le \mu - (\sigma/\sqrt{2})]$. Define*

$$\gamma := \min\left(2\delta - 1 + \frac{1}{\sqrt{2}}\left(\frac{\sigma}{2^{k-1}}\right), 2^{k-K-1}\right).$$

Then,

$$\mathbf{P}\left[\varepsilon \ge (1+\gamma)2^{k-1}\right] \ge \frac{1}{4}\left(\frac{\sigma}{(1-\delta)2^k}\right)^2.$$

Proof. If $\gamma \le 2\delta - 1 + (1/\sqrt{2})\left(\sigma/2^{k-1}\right)$, we obtain

$$\mathbf{P}\left[\varepsilon \ge (1+\gamma)2^{k-1}\right] \ge \mathbf{P}\left[\varepsilon \ge \mu + \frac{\sigma}{\sqrt{2}}\right] \ge \frac{1}{2} \cdot \mathbf{P}\left[|\varepsilon - \mu| \ge \frac{\sigma}{\sqrt{2}}\right],$$

where the last inequality holds because $\mathbf{P}[\varepsilon \ge \mu + (\sigma/\sqrt{2})] \ge \mathbf{P}[\varepsilon \le \mu - (\sigma/\sqrt{2})]$. Since $2^k - \mu \le (1-\delta)2^k$, we obtain from Lemma B.5

$$\mathbf{P}\left[\varepsilon \ge (1+\gamma)2^{k-1}\right] \ge \frac{1}{2} \cdot \frac{\sigma^2 - (1/2)\sigma^2}{((1-\delta)2^k)^2} = \frac{1}{4}\left(\frac{\sigma}{(1-\delta)2^k}\right)^2. \quad \square$$

Note that we have to make sure that $\gamma > 0$. Therefore, for $\delta < \frac{1}{2}$, the definition of γ in Lemma B.6 makes sense only if we require $(\sigma/2^{k-1}) > (1-2\delta)\cdot\sqrt{2}$.

Corollary 2. *If f is a well-shaped distribution, we have $\delta = \frac{1}{2}$ and thus,*

$$\alpha = \left(\frac{\sigma}{2^k}\right)^2, \quad \textit{where } \gamma = \min\left(\frac{1}{\sqrt{2}}\left(\frac{\sigma}{2^{k-1}}\right), 2^{k-K-1}\right).$$

We come to property (P2). The next lemma characterizes distributions that satisfy (P2).

Lemma B.7. *Let ε be a random variable chosen according to a distribution f over $[1, 2^k]$. Let l be some integer, $0 \le l \le k$, such that for each i, $0 \le i \le k-l$, $\mathbf{P}[\varepsilon \le 2^i] \le 2^i \cdot (1/2)^{k-l}$. Then, $\sum_{i=0}^{k} \mathbf{P}[\varepsilon \le 2^i] \le 2 + l$.*

Proof.

$$\sum_{i=0}^{k}\mathbf{P}\left[\varepsilon \le 2^i\right] = \sum_{i=0}^{k-l}\mathbf{P}\left[\varepsilon \le 2^i\right] + \sum_{i=k-l+1}^{k}\mathbf{P}\left[\varepsilon \le 2^i\right] \le \sum_{i=0}^{k-l}\left(\frac{1}{2}\right)^{k-l-i} + \sum_{i=k-l+1}^{k} 1 = \sum_{i=0}^{k-l}\left(\frac{1}{2}\right)^{i} + l \le 2 + l. \quad \square$$

Corollary 3. *If f is a well-shaped distribution, then $\beta = 2$.*

Proof. Since f is nondecreasing in $[1, 2^{k-1}]$, the distribution function $F(x) := \mathbf{P}[\varepsilon \le x]$ of f is *strictly* increasing once $F(x) > 0$. Moreover, since f is symmetric around μ and $\mu \ge 2^{k-1}$, $F(2^{k-1}) \le \frac{1}{2}$. Thus, $F(2^i) \le 2^i \cdot (1/2)^k$ for each i, $0 \le i \le k-1$. Clearly, $F(2^k) \le 1$. □

Finally, consider property (P3). We remark that $\mathbf{P}\left[\varepsilon \ge (1+\gamma)2^{k-1}\right] \ge \alpha$ implies $\mathbf{E}[\varepsilon] \ge \frac{1}{2}(1+\gamma)\alpha 2^k$. However, this bound on δ might be too weak. In Lemma B.7, we require $\mathbf{P}[\varepsilon \le x] \le x \cdot (1/2)^{k-l}$ only for each $x = 2^i$, where $0 \le i \le k-l$. If we instead require that this relation holds for every $x \in [1, 2^{k-l}]$, we obtain a characterization for (P3).

Lemma B.8. *Let ε be a random variable chosen according to a distribution f over $[1, 2^k]$. Let l be some integer, $0 \le l \le k$, such that for each $x \in [1, 2^{k-l}]$, $\mathbf{P}[\varepsilon \le x] \le x \cdot (1/2)^{k-l}$. Then, $\mathbf{E}[\varepsilon] \ge (1/2^{l+1}) \cdot 2^k$.*

Proof. Consider a uniform random variable U over $[1, 2^{k-l}]$. We have $G(x) := \mathbf{P}[U \le x] = \min(x \cdot (1/2)^{k-l}, 1)$; see also Figure 2. By definition, $\mathbf{P}[\varepsilon > x] \ge \mathbf{P}[U > x]$ for each $x \in [1, 2^k]$. That is, ε stochastically dominates U, and therefore $\mathbf{E}[\varepsilon] \ge \mathbf{E}[U] = (2^{k-l}+1)/2$. □

For example, well-shaped distributions satisfy Lemma B.8 with $l = 1$, which yields $\mathbf{E}[\varepsilon] \ge \frac{1}{4} \cdot 2^k$.

Appendix C. Proof of Lemma 5.1. We introduce some additional notation. The volume $V^{\mathcal{S}}(t)$ is the sum of the remaining processing times of the jobs that are active at time t. $L^{\mathcal{S}}(t)$ denotes the total work done prior to time t, i.e., the overall time the machine has been processing jobs until time t. For a generic function ϑ ($= \delta$, V, or L), we define $\Delta\vartheta(t) := \vartheta^{\text{MLF}}(t) - \vartheta^{\text{OPT}}(t)$. For ϑ ($= \delta$, V, ΔV, L, or ΔL), the notation $\vartheta_{=k}(t)$ will denote the value of ϑ at time t when restricted to jobs of class k. We use $\vartheta_{\ge h, \le k}(t)$ to denote the value of ϑ at time t when restricted to jobs of classes between h and k.

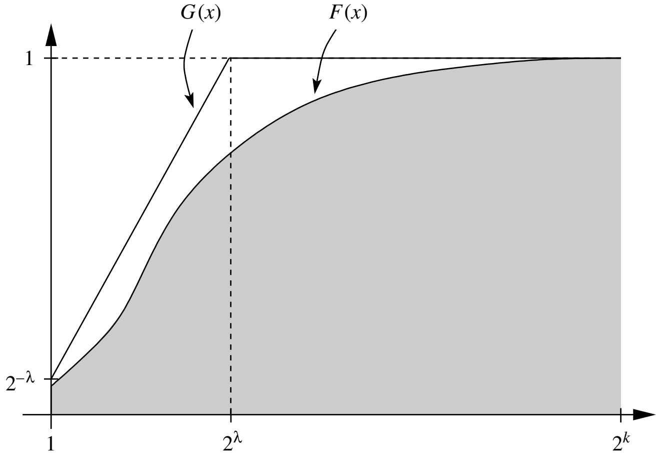

Figure 2. $F(x) := \mathbf{P}[\varepsilon \le x]$, $G(x) := \min(x \cdot (1/2)^{\lambda}, 1)$, where $\lambda := k - l$.

LEMMA 5.1. *For any input instance I, at any time t, $\delta_I^l(t) \le h_I(t) + (6/\gamma)\delta_I^{\text{OPT}}(t)$.*

PROOF. In the following, we omit I when clear from the context. Denote by k_1 and k_2, respectively, the lowest and highest class such that at least one job of that class is in the system at time t. We bound the number of lucky jobs that are active at time t as follows:

$$\delta^l(t) \le h(t) + \frac{1}{\gamma}\sum_{i=k_1}^{k_2} \frac{V_{=i}^{\text{MLF}}(t)}{2^{i-1}}. \tag{8}$$

The bound follows since every job that is lucky at time t is either an ending head job or not. An ending head job might have been processed, and therefore we cannot assume anything about its remaining processing time. However, the number of ending head jobs is $h(t)$. For all other lucky jobs, we can bound the remaining processing time from below: A job of class i has remaining processing time at least $\gamma 2^{i-1}$. We have

$$\begin{aligned}
\sum_{i=k_1}^{k_2} \frac{V_{=i}^{\text{MLF}}(t)}{2^{i-1}} &= \sum_{i=k_1}^{k_2} \frac{V_{=i}^{\text{OPT}}(t) + \Delta V_{=i}(t)}{2^{i-1}} \\
&\le 2\delta_{\ge k_1,\, \le k_2}^{\text{OPT}}(t) + \sum_{i=k_1}^{k_2} \frac{\Delta V_{=i}(t)}{2^{i-1}} \\
&= 2\delta_{\ge k_1,\, \le k_2}^{\text{OPT}}(t) + 2\sum_{i=k_1}^{k_2} \frac{\Delta V_{\le i}(t) - \Delta V_{\le i-1}(t)}{2^i} \\
&= 2\delta_{\ge k_1,\, \le k_2}^{\text{OPT}}(t) + 2\frac{\Delta V_{\le k_2}(t)}{2^{k_2}} + 2\sum_{i=k_1}^{k_2-1} \frac{\Delta V_{\le i}(t)}{2^{i+1}} \\
&\le 2\delta_{\ge k_1,\, \le k_2}^{\text{OPT}}(t) + \delta_{\le k_1-1}^{\text{OPT}}(t) + 4\sum_{i=k_1}^{k_2} \frac{\Delta V_{\le i}(t)}{2^{i+1}} \\
&\le 2\delta_{\le k_2}^{\text{OPT}}(t) + 4\sum_{i=k_1}^{k_2} \frac{\Delta V_{\le i}(t)}{2^{i+1}},
\end{aligned} \tag{9}$$

where the second inequality follows since a job of class i has size at most 2^i, while the fourth inequality follows since $\Delta V_{\le k_1 - 1}(t) = 0$ by definition.

We are left to study the sum in (9). For any $t_1 \le t_2 \le t$ and a generic function ϑ, denote by $\vartheta^{[t_1,t_2]}(t)$ the value of ϑ at time t when restricted to jobs released between t_1 and t_2, e.g., $L_{\le i}^{[t_1,t_2]}(t)$ is the work done by time t on jobs of class at most i released between time t_1 and t_2. Denote by $t_i < t$ the maximum between 0 and the last time prior to time t in which a job was processed in queue Q_{i+1} or higher in this specific execution of MLF. Observe that, for $i = k_1, \dots, k_2$, $[t_{i+1}, t) \supseteq [t_i, t)$.

At time t_i, either the algorithm was processing a job in queue Q_{i+1} or higher, or $t_i = 0$. Thus, at time t_i no jobs were in queues $Q_0, \dots, Q_i$. Therefore,

$$\Delta V_{\le i}(t) \le \Delta V_{\le i}^{(t_i,t]}(t) \le L_{>i}^{\text{MLF}(t_i,t]}(t) - L_{>i}^{\text{OPT}(t_i,t]}(t) = \Delta L_{>i}^{(t_i,t]}(t).$$

In the following, we adopt the convention $t_{k_1-1} := t$. From the above, we have

$$\begin{aligned}
\sum_{i=k_1}^{k_2} \frac{\Delta L_{>i}^{(t_i,t]}(t)}{2^{i+1}} &= \sum_{i=k_1}^{k_2} \frac{L_{>i}^{\text{MLF}(t_i,\, t]}(t) - L_{>i}^{\text{OPT}(t_i,\, t]}(t)}{2^{i+1}} \\
&= \sum_{i=k_1}^{k_2} \sum_{j=k_1-1}^{i-1} \frac{L_{>i}^{\text{MLF}(t_{j+1},\, t_j]}(t) - L_{>i}^{\text{OPT}(t_{j+1},\, t_j]}(t)}{2^{i+1}} \\
&= \sum_{j=k_1-1}^{k_2-1} \sum_{i=j+1}^{k_2} \frac{L_{>i}^{\text{MLF}(t_{j+1},\, t_j]}(t) - L_{>i}^{\text{OPT}(t_{j+1},\, t_j]}(t)}{2^{i+1}},
\end{aligned}$$

where the second equality follows by partitioning the work done on the jobs released in the interval $(t_i, t]$ into the work done on the jobs released in the intervals $(t_{j+1}, t_j]$, $j = k_1 - 1, \dots, i - 1$.

Let $\bar{i}(j) \in \{j+1, \dots, k_2\}$ be the index that maximizes $L_{>i}^{\text{MLF}(t_{j+1},t_j]} - L_{>i}^{\text{OPT}(t_{j+1},t_j]}$. Then,

$$\begin{aligned}\sum_{i=k_1}^{k_2} \frac{\Delta L_{>i}^{(t_i,t]}(t)}{2^{i+1}} &\le \sum_{j=k_1-1}^{k_2-1} \sum_{i=j+1}^{k_2} \frac{L_{>\bar{i}(j)}^{\text{MLF}(t_{j+1},t_j]}(t) - L_{>\bar{i}(j)}^{\text{OPT}(t_{j+1},t_j]}(t)}{2^{i+1}}\\ &\le \sum_{j=k_1-1}^{k_2-1} \frac{L_{>\bar{i}(j)}^{\text{MLF}(t_{j+1},t_j]}(t) - L_{>\bar{i}(j)}^{\text{OPT}(t_{j+1},t_j]}(t)}{2^{j+1}}\\ &\le \sum_{j=k_1-1}^{k_2-1} \delta_{>\bar{i}(j)}^{\text{OPT}(t_{j+1},t_j]}(t) \le \delta_{\ge k_1}^{\text{OPT}(t_{k_2},t]}(t) \le \delta_{\ge k_1}^{\text{OPT}}(t).\end{aligned}$$

To prove the third inequality, observe that every job of class larger than $\bar{i}(j) > j$ released in the time interval $(t_{j+1}, t_j]$ is processed by MLF in the interval $(t_{j+1}, t]$ for at most 2^{j+1} time units. Order the jobs of this specific set by increasing $x_j^{\text{MLF}}(t)$. Now, observe that each of these jobs has initial processing time at least $2^{\bar{i}(j)} \ge 2^{j+1}$ at their release, and we give to the optimum the further advantage that it finishes every such job when processed for an amount $x_j^{\text{MLF}}(t) \le 2^{j+1}$. To maximize the number of finished jobs, the optimum places the work $L_{>\bar{i}(j)}^{\text{OPT}(t_{j+1},t_j]}$ on the jobs with smaller $x_j^{\text{MLF}}(t)$. The optimum is then left at time t with a number of jobs

$$\delta_{>\bar{i}(j)}^{\text{OPT}(t_{j+1},t_j]}(t) \ge \frac{L_{>\bar{i}(j)}^{\text{MLF}(t_{j+1},t_j]}(t) - L_{>\bar{i}(j)}^{\text{OPT}(t_{j+1},t_j]}(t)}{2^{j+1}}.$$

Altogether, we obtain from (8), (9), and (10)

$$\delta^l(t) \le h(t) + \frac{2}{\gamma}\delta_{\le k_2}^{\text{OPT}}(t) + \frac{4}{\gamma}\delta_{\ge k_1}^{\text{OPT}}(t) \le h(t) + \frac{6}{\gamma}\delta^{\text{OPT}}(t). \quad \square$$

Appendix D. Proving positive and negative correlations. In Lemmas 5.2 and 5.9, we use a technique described in the book by Alon and Spencer [1, Chapter 6] to prove that two events are negatively or positively correlated. Two events A and B are *positively correlated* if $\mathbf{P}[A \cap B] \ge \mathbf{P}[A]\,\mathbf{P}[B]$, while A and B are *negatively correlated* if $\mathbf{P}[A \cap B] \le \mathbf{P}[A]\,\mathbf{P}[B]$. We give some more details in this section.

Let Ω denote a finite probability space with probability function $\mathbf{P}$. Let A and B denote two events in Ω. A and B are positively or negatively correlated if the following three conditions hold:

(i) Ω forms a *distributive lattice*. A lattice $(\Omega, \le, \vee, \wedge)$ is a partially ordered set $(\Omega, \le)$ in which every two elements x and y have a unique minimal upper bound, denoted by $x \vee y$, and a unique maximal lower bound, denoted by $x \wedge y$. A lattice $(\Omega, \le, \vee, \wedge)$ is distributive if for all $x, y, z \in \Omega$: $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.

(ii) The probability function $\mathbf{P}$ is *log supermodular*, i.e., for all $x, y \in \Omega$,

$$\mathbf{P}[x] \cdot \mathbf{P}[y] \le \mathbf{P}[x \vee y] \cdot \mathbf{P}[x \wedge y].$$

(iii) An event $E \subseteq \Omega$ is *monotone increasing* if $x \in E$ and $x \le y$ implies that $y \in E$, while $E \subseteq \Omega$ is *monotone decreasing* if $x \in E$ and $x \ge y$ implies that $y \in E$. A and B are positively correlated if both A and B are monotone increasing or monotone decreasing. A and B are negatively correlated if A is monotone decreasing and B is monotone increasing or vice versa.

In both Lemma 5.2 and Lemma 5.9, we need to prove that two events A' and B' are correlated; in Lemma 5.2, $A' := (p_{q_i} \le 2^i)$ and $B' := (H(t) = H)$, and in Lemma 5.9, $A' := (\varepsilon_j \ge \gamma 2^{i-1})$ and $B' := (p_j > \pi_j)$. In both cases, A' is an event that solely depends on the perturbation of some job j, e.g., $j := q_i$ in Lemma 5.2 and j itself in Lemma 5.9. We condition the probability space in order to make sure that only the processing time of j is random. That is, we fix the processing times of all jobs other than j to $\mathbf{x}_{\bar{j}}$, which we denote by $(\mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}})$. Define $A := (A' \mid \mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}})$ and $B := (B' \mid \mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}})$. Let Ω denote the conditioned probability space and let $\mathbf{P}$ denote the underlying conditioned probability distribution. The following two statements are easy to verify.

(i) Ω together with the partial order $\le$ and the standard max and min operations constitutes a distributive lattice.

(ii) $\mathbf{P}$ is log supermodular. The inequality holds even with equality and does not depend on the underlying probability distribution.

We next argue that the events A and B are monotone increasing or decreasing.

Addition to Proof of Lemma 5.2. Let the processing time p_{jI} of job $j = q_i$ in I be fixed such that $I \in A = (p_{q_i} \leq 2^i \mid \mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}})$. Define an instance I' with $p_{jI'} \leq p_{jI}$. Then, $I' \in A$. Hence, A is monotone decreasing. On the other hand, if the processing time p_{jI} in I is chosen such that $I \in B = (H(t) = H \mid \mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}})$, i.e., j is a head job at time t, then j remains a head job in any instance I' with $p_{jI'} \geq p_{jI}$. Therefore, B is monotone increasing. From the discussion above, we conclude that A and B are negatively correlated.

Addition to Proof of Lemma 5.9. Let I be an instance with processing time p_{jI} of j being such that $I \in A = (\varepsilon_j \geq \gamma 2^{i-1} \mid \mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}})$. Consider an instance I' with processing time $p_{jI'} \geq p_{jI}$. Clearly, $I' \in A$ and thus, A is monotone increasing. Similarly, let p_{jI} be fixed such that $I \in B = (p_j > \pi_j \mid \mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}})$. If we consider an instance I' with $p_{jI'} \geq p_{jI}$, then j also satisfies $(p_{jI'} > \pi_j)$ and thus, $I' \in B$. That is, B is monotone increasing. We conclude that A and B are positively correlated.

Since the processing times of all jobs are perturbed independently, A' and $(\mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}})$ are independent, i.e., $\mathbf{P}\left[A' \mid \mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}}\right] = \mathbf{P}[A']$. We exploit this fact as follows in order to prove that the events A' and B' are also correlated the second inequality is due to the correlation of A and B:

$$\begin{aligned}\mathbf{P}[A' \cap B'] &= \sum_{\mathbf{x}_{\bar{j}}} \mathbf{P}\left[A' \cap B' \mid \mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}}\right] \mathbf{P}\left[\mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}}\right] \\ &\lesseqgtr \sum_{\mathbf{x}_{\bar{j}}} \mathbf{P}\left[A' \mid \mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}}\right] \mathbf{P}\left[B' \mid \mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}}\right] \mathbf{P}\left[\mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}}\right] \\ &= \mathbf{P}[A'] \sum_{\mathbf{x}_{\bar{j}}} \mathbf{P}\left[B' \mid \mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}}\right] \mathbf{P}\left[\mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}}\right] = \mathbf{P}[A']\,\mathbf{P}[B'].\end{aligned}$$

The above reasoning clearly holds for the oblivious adversary. Observe, however, that it also holds in the adaptive case: The event A' only depends on the random outcome ε_j of job j, which the adaptive adversary cannot control. In principle, the event B' might be influenced by a change in the processing time of j. However, since p_j is increased in both cases, this change is revealed to the adversary only after the completion of j itself. So, up to time t, the behavior of the adaptive adversary will be the same.

Acknowledgments. The authors thank Alessandro Panconesi, Kirk Pruhs, and Gerhard Woeginger for helpful discussions and suggestions. Moreover, we are grateful to two anonymous referees whose valuable comments helped to improve the exposition of the paper.

The authors were partially supported by the EU Project AMORE Grant HPRN-CT-1999-00104, IST Programme of the EU under Contract IST-1999-14186 (ALCOM-FT), IST-2000-14084 (APPOL), IST-2001-33555 (COSIN), MIUR Project "Resource Allocation in Real Networks," DFG Research Center "Mathematics for Key Technologies" (FZT 86) in Berlin and by the DFG Graduated Studies Program "Quality Guarantees for Computer Systems." This work was done while the fourth and the fifth author were visitors at Università di Roma "La Sapienza," Rome, Italy. An extended abstract of this paper appeared in the Conference Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, 2003 (Becchetti et al. [4]).

References

[1] Alon, N., J. H. Spencer. 2000. *The Probabilistic Method*, 2nd ed. Wiley-Interscience Series in Discrete Mathematics. Wiley, Chichester, UK.

[2] Banderier, C., R. Beier, K. Mehlhorn. 2003. Smoothed analysis of three combinatorial problems. *28th Internat. Sympos. Math. Foundations Comput. Sci.* LNCS, No. 2747. Springer, Berlin, Germany, 198–207.

[3] Becchetti, L., S. Leonardi. 2004. Nonclairvoyant scheduling to minimize the total flow time on single and parallel machines. *J. ACM* **51**(4) 517–539.

[4] Becchetti, L., S. Leonardi, A. Marchetti-Spaccamela, G. Schäfer, T. Vredeveld. 2003. Average case and smoothed competitive analysis of the multi-level feedback algorithm. *Proc. 44th Annual IEEE Sympos. Foundations Comput. Sci.* 462–471.

[5] Beier, R., B. Vöcking. 2003. Random knapsack in expected polynomial time. *Proc. 35th Annual ACM Sympos. Theory Comput.* 232–241.

[6] Beier, R., B. Vöcking. 2004. Typical properties of winners and losers in discrete optimization. *Proc. 36th Annual ACM Sympos. Theory Comput.* 343–352.

[7] Beier, R., A. Czumaj, P. Krysta, B. Vöcking. 2004. Computing equilibria for congestion games with (im)perfect information. *Proc. 15th Annual ACM-SIAM Sympos. Discrete Algorithms* 739–748.

[8] Blum, A., J. Dunagan. 2002. Smoothed analysis of the perceptron algorithm. *Proc. 13th Annual ACM-SIAM Sympos. Discrete Algorithms* 905–914.

[9] Borodin, A., R. El-Yaniv. 1998. *Online Computation and Competitive Analysis*. Cambridge University Press, New York.

[10] Borodin, A., S. Irani, P. Raghavan, B. Schieber. 1995. Competitive paging with locality of reference. *J. Comput. System Sci.* **50**(2) 244–258.

[11] Feller, W. 1967. *An Introduction to Probability Theory and Its Applications*. John Wiley & Sons, Inc., New York.

[12] Grimmett, G. R., D. R. Stirzaker. 2001. *Probability and Random Processes*, 3rd ed. Oxford University Press, New York.

[13] Kalyanasundaram, B., K. Pruhs. 2000. Speed is as powerful as clairvoyance. *J. ACM* **47**(4) 617–643.

[14] Kalyanasundaram, B., K. Pruhs. 2003. Minimizing flow time nonclairvoyantly. *J. ACM* **50**(4) 551–567.

[15] Koutsoupias, E., C. Papadimitriou. 1994. Beyond competitive analysis. *Proc. 25th Sympos. Foundations Comput. Sci.* 394–400.

[16] Leonardi, S., D. Raz. 1997. Approximating total flow time on parallel machines. *Proc. 29th Annual ACM Sympos. Theory Comput.* 110–119.

[17] Michel, J. E., E. G. Coffman, Jr. 1974. Synthesis of a feedback queueing discipline for computer operation. *J. ACM* **21**(2) 329–339.

[18] Motwani, R., P. Raghavan. 1995. *Randomized Algorithms*, 1st ed. Cambridge University Press, Cambridge, UK.

[19] Motwani, R., S. Phillips, E. Torng. 1994. Non-clairvoyant scheduling. *Theoretical Comput. Sci.* **130** 17–47.

[20] Nutt, G. 1999. *Operating System Projects Using Windows NT*. Addison-Wesley, Reading, MA.

[21] Scharbrodt, M., T. Schickinger, A. Steger. 2002. A new average case analysis for completion time scheduling. *Proc. 34th Annual ACM Sympos. Theory Comput.*, 170–178.

[22] Schrage, L. 1968. A proof of the optimality of the shortest remaining processing time discipline. *Oper. Res.* **16** 687–690.

[23] Sleator, D., R. E. Tarjan. 1985. Amortized efficiency of list update and paging rules. *Comm. ACM* **28** 202–208.

[24] Spielman, D., S. H. Teng. 2003. Smoothed analysis of termination of linear programming algorithms. *Math. Programming* **97**(1–2) 375–404.

[25] Spielman, D. A., S. H. Teng. 2004. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM* **51**(3) 385–463.

[26] Tanenbaum, A. S. 1992. *Modern Operating Systems*. Prentice-Hall Inc., Upper Saddle River, NJ.