I N F O R M A T I K

Average Case and Smoothed Competitive Analysis of the Multi-Level Feedback Algorithm

Luca Becchetti
Stefano Leonardi
Alberto Marchetti-Spaccamela
Guido Schäfer
Tjark Vredeveld

MPI–I–2003–1–014 July 2003

FORSCHUNGSBERICHT RESEARCH REPORT

M A X - P L A N C K - I N S T I T U T
F Ü R
I N F O R M A T I K

Stuhlsatzenhausweg 85 66123 Saarbrücken Germany

Authors' Addresses

First three authors:
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00196 Roma, Italy
Email: {becchett,leon,alberto}@dis.uniroma1.it

Fourth author:
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
Email: schaefer@mpi-sb.mpg.de

Fifth author:
Konrad-Zuse-Zentrum für Informationstechnik
Takustraße 7, 14195 Berlin, Germany
Email: vredeveld@zib.de

Publication Notes

An extended abstract of this paper was accepted for publication in the proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, 2003.

Acknowledgements

Partially supported by the EU project AMORE grant HPRN-CT-1999-00104, IST Programme of the EU under contract number 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 DFG graduated studies program "Quality Guarantees for Computer Systems" (GK 623), Saarbrücken. This work was done while the fourth and the fifth author were visitors at Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Italy.

Abstract

In this paper we introduce the notion of smoothed competitive analysis of online algorithms. Smoothed analysis has been proposed by Spielman and Teng [23] to explain the behaviour 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 Multi-Level Feedback (MLF) algorithm 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, where the initial processing times are smoothened 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 smoothened according to the partial bit randomization model. For various other smoothening models, including the additive symmetric smoothening model used by Spielman and Teng [23], 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 distribution.

Keywords

Smoothed analysis, average case analysis, online algorithms, non-clairvoyant scheduling, Mutli-Level Feedback algorithm, average flow time.

1 Introduction

Smoothed analysis was proposed by Spielman and Teng [23] 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 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 $\times$ running time) space. Spielman and Teng's striking result 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 [6, 10, 20, 25, 24], smoothed analysis was successfully applied to characterize the time complexity of other problems.

Competitive analysis [22] 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 over-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 [8] and the resource augmentation model for analyzing online scheduling algorithms [13]. More related to our work is the *diffuse adversary model* of Koutsoupias and Papadimitriou [14], a refinement of competitive analysis that assumes that the actual distribution of the input is a member of a known class of possible distributions chosen by a worst case adversary.

Smoothed Competitive Analysis. In this paper we introduce the notion of *smoothed competitiveness*. The competitive ratio c of an online deterministic algorithm $\mathcal{A}$ 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., $c = \sup_{\bar{I}}(\mathcal{A}_{\bar{I}}/\mathcal{OPT}_{\bar{I}})$. Following the idea of Spielman and Teng [23], we smoothen the input instance according to some probability distribution f. We define the *smoothed competitive ratio* as

$$c = \sup_{\bar{I}} \mathbf{E}_{I \in_f N(\bar{I})} \left[\frac{\mathcal{A}_I}{\mathcal{OPT}_I} \right],$$

where the supremum is taken over all input instances $\bar{I}$, and the expectation is taken over all instances I that are obtainable by smoothening the input instance $\bar{I}$ according to f in the neighborhood $N(\bar{I})$. Observe that we might alternatively define the smoothed competitive ratio as the ratio of the expectations in the expression above. We also address this issue in the paper.

This kind of analysis results in having the algorithm and the smoothening 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 to prove upper and lower bounds against different adversaries.

In a way similar to the analysis of randomized online algorithms [7], 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 smoothening 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 input instance. Considering the instance space, in the oblivious case $N(\bar{I})$ is defined at the beginning, once the adversary has fixed $\bar{I}$, while in the adaptive case $N(\bar{I})$ is itself a random variable, since it depends 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.

The Multi-Level Feedback Algorithm. One of the most successful online algorithms used in practice is the Multi-Level Feedback algorithm (MLF) for processor scheduling in a time sharing multitasking operating system. MLF is a *non-clairvoyant* scheduling algorithm, i.e., scheduling decisions are taken without knowledge of the time a job needs to be executed. Windows NT [19] and Unix [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 effective in practice, it behaves poorly with respect to worst case analysis. Assuming that processing times are chosen in $[1, 2^K]$, Motwani et al. [17] showed a lower bound of $\Omega(2^K)$ for any deterministic non-clairvoyant preemptive scheduling algorithm. The next step was then to use randomization. A randomized version of the Multi-Level Feedback algorithm (RMLF) was first proposed by Kalyanasundaram and Pruhs [12] for a single machine achieving an $O(\log n \log\log n)$ competitive ratio against the online adaptive adversary, where n is the number of jobs that are released. Becchetti and Leonardi present a version of RMLF achieving an $O(\log n \log \frac{n}{m})$ 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 [17].

Contribution of this Paper. In this paper, we apply smoothed competitive analysis to the Multi-Level Feedback algorithm. For smoothening 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 $[1, 2^k]$. A similar model was used by Beier et al. [5] and Banderier et al. [2]. 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. In [5] and [2] only the uniform distribution was considered. 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.

    We have defined the smoothed competitive ratio as the supremum, over the set of possible input instances, of the expected ratio between the cost of the algorithm and the optimal cost. An alternative is to define it as the ratio between the expected costs of the algorithm and of the optimum, see also [21]. 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 distribution. 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 [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 much stronger than average completion time. Early work by Michel and Coffman [16] only considered the problem of synthesizing a feedback queue system under Poisson arrivals and a known discrete probability distribution on processing times so that pre-specified 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 $k \leq K/3$, against an oblivious adversary for any deterministic algorithm when run on processing times smoothened according to the partial bit randomization model.

(iv) Spielman and Teng [23] used an additive symmetric smoothening model, where each input parameter is smoothened symmetrically around its initial value. A natural question is whether 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 smoothening models, including the additive symmetric, the additive relative symmetric and the multiplicative smoothening model.

2 Problem Definition and Smoothening Models

The adversary releases a set $J = \{1, \ldots, n\}$ of n jobs over time. Each job j has a release time r_j and an initial processing time $\bar{p}_j$. We assume that the initial processing times are integers in $[1, 2^K]$. We allow preemption of jobs, i.e., a job that is running can be interrupted and resumed later on the machine. The algorithm decides which uncompleted job should be executed at each time. The machine can process at most one job at a time and a job cannot be processed before its release time. For a generic schedule $\mathcal{S}$, let $C_j^{\mathcal{S}}$ denote the completion time of job j. Then, 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 given by $F^{\mathcal{S}} = \sum_{j \in J} F_j^{\mathcal{S}}$. A *non-clairvoyant* 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 total flow time.

The input instance may be smoothened according to different smoothening models. We discuss four different smoothening models below. We only smoothen the processing times of the jobs. One could additionally smoothen the release dates. However, for our analysis to hold it is sufficient to smoothen the processing times only. 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.

Additive Symmetric Smoothening Model. In the additive symmetric smoothening model the processing time of each job is smoothened symmetrically around its initial processing time. The smoothed processing time 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 [23].

$$p_j = \max(1, \bar{p}_j + \epsilon_j), \text{ where } \epsilon_j \overset{f}{\leftarrow} [-L, L].$$

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

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

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

Multiplicative Smoothening Model. In the multiplicative smoothening model the processing time of each job is smoothened symmetrically around its initial processing time. The smoothed processing times are chosen independently according to f from the the range $[(1-\epsilon)\bar{p}_j, (1+\epsilon)\bar{p}_j]$ for some $\epsilon > 0$. This model is also discussed but not analyzed by Spielman and Teng [23].

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

Partial Bit Randomization Model. The initial processing times are smoothened 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

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

Note that ϵ_j is at least 1 and therefore 1 is subtracted from $\bar{p}_j$ before applying the modification. For $k = 0$, this assures that the smoothed processing times are equal to the initial processing times. For $k = K$, the processing times are randomly chosen from $[1, 2^K]$ according to the underlying distribution. A similar model is used by Beier et al. [5] and Banderier et al. [2].

As will be seen later, MLF is not competitive at all under any of the first three models: MLF may admit a smoothed competitive ratio of $\Omega(2^K)$. Therefore, these models are not suitable to explain the success of MLF in practice. The model we use is the partial bit randomization model.

Our analysis holds for any *well-shaped* distribution f over $[1, 2^k]$. A probability density function f is well-shaped if it satisfies the following conditions:

(i) f is symmetric around its mean,

(ii) the mean μ of f is centered in $[1, 2^k]$, and

(iii) f is non-decreasing in the range $[1, \mu]$.

In the sequel, we denote by σ the standard deviation of f. We emphasize that the distribution may be discrete as well as continuous.

We discuss some features of the smoothed processing times. Let ϕ_j be defined as $\phi_j = 2^k \lfloor \frac{\bar{p}_j - 1}{2^k} \rfloor$. Then, $p_j = \phi_j + \epsilon_j$. Consider a job j with initial processing time in $[1, 2^k]$. Then, the initial processing time of j is completely replaced by some random processing time in $[1, 2^k]$ chosen according to the probability distribution f.

Fact 1. *Let $\bar{p}_j \in [1, 2^k]$. Then, $\phi_j = 0$ and thus $p_j \in [1, 2^k]$. Moreover, $\mathbf{P}[p_j \leq x] = \mathbf{P}[\epsilon_j \leq x]$ for each $x \in [1, 2^k]$.*

Next, consider a job j with initial processing time $\bar{p}_j \in (2^{i-1}, 2^i]$, for some $i > k$. 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 2. *Let $\bar{p}_j \in (2^{i-1}, 2^i]$ for some i, $k < i \leq K$. Then, $2^{i-1} \leq \phi_j \leq 2^i - 2^k$ and thus $p_j \in (2^{i-1}, 2^i]$.*

3 The Multi-Level Feedback Algorithm

In this section we describe the Multi-Level Feedback (MLF) algorithm. We say that a job is *alive* or *active* at time t in a schedule $\mathcal{S}$, if it has been released but not completed at this time, i.e., $r_j \leq t < C_j^{\mathcal{S}}$. 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 $\mathcal{A}$ the schedule produced by MLF.

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.

1. Job j released at time t enters queue Q_0.

2. Schedule on the machine the alive job that has highest priority in the lowest non-empty queue.
3. For a job j in a queue Q_i at time t, if $x_j^{\mathcal{A}}(t) = p_j$, assign $C_j^{\mathcal{A}} = t$ and remove the job from the queue.
4. For a job j in a queue Q_i at time t, if $x_j^{\mathcal{A}}(t) = 2^i < p_j$, job j is moved from Q_i to Q_{i+1}.

4 Smoothed Analysis

4.1 Preliminaries

We classify jobs into classes according to their processing times: a job j is of *class* $i \geq 0$, if $p_j \in (2^{i-1}, 2^i]$. Observe that a job of class i will end in queue Q_i. Since all processing times are in $[1, 2^K]$, the maximum class of a job is K. Moreover, during the execution of the algorithm at most $K+1$ queues are created. We denote by $\delta^{\mathcal{S}}(t)$ the number of jobs that are active at time t in $\mathcal{S}$. We use $S^{\mathcal{S}}(t)$ to refer to the set of active jobs at time t. We use $\mathcal{A}$ and $\mathcal{OPT}$ to denote the schedule produced by MLF and by an optimal algorithm, respectively. We state the following facts.

Fact 3 ([15]). $F^{\mathcal{S}} = \sum_{j \in J} F_j^{\mathcal{S}} = \int_t \delta^{\mathcal{S}}(t)dt.$

Fact 4. $F^{\mathcal{S}} \geq \sum_{j \in J} p_j.$

Fact 5. *At any time t and for any i, at most one job, alive at time t, has been executed in queue Q_i but has not been promoted to Q_{i+1}.*

A lucky job is a job that still has a reasonably large remaining processing time when it enters its final queue. More precisely, a job j of class i is called *lucky* if $p_j - 2^{i-1} \geq \gamma_k 2^{i-1}$; otherwise, it is called *unlucky*. Here, γ_k depends on k and the standard deviation σ of the distribution and is defined as $\gamma_k = \min(\frac{1}{\sqrt{2}}(\frac{\sigma}{2^{k-1}}), 2^{k-K})$. We use β_k to refer to the fraction $1/\gamma_k$. We use $\delta^l(t)$ to denote the number of lucky jobs that are active at time t in MLF. 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 head jobs that are ending.

We define the following random variables. For each job j, X_j^l has value 1 if job j is lucky, while $X_j^l = 0$ if j is unlucky. We use $Cl_j \in [0, k]$ to denote the class of a job j. Note that the class of a job with $\bar{p}_j \in (2^{i-1}, 2^i]$, for some $i > k$, is not a random variable. Moreover, for each job j and for each time t, two binary variables are defined: $X_j(t)$ and $X_j^l(t)$. The value of $X_j(t)$ is 1 if job j is alive at time t, and 0 otherwise. $X_j^l(t)$ is defined in terms of X_j^l and $X_j(t)$, namely, $X_j^l(t) = X_j^l \cdot X_j(t)$.

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

We prove our main result in Subsection 4.2. The proof uses a high probability argument which, for the sake of clarity, is given in Subsection 4.3. Due to lack of space most of the proofs are only sketched. The complete proofs are given in the appendix.

4.2 Smoothed Competitiveness of MLF

In this section we prove that MLF is $O((2^k/\sigma)^3 + (2^k/\sigma)^2 2^{K-k})$-competitive.

Lemma 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 in [3].

Lemma 1. *For any input instance I, at any time t, $\delta_I^l(t) \le h_I(t) + 6\beta_k \delta_I^{\mathcal{OPT}}(t)$.*

In the sequel, we exploit the fact that two events A and B are correlated: 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 $\mathbf{P}[A \cap B] \le \mathbf{P}[A]\,\mathbf{P}[B]$. In the book by Alon and Spencer [1, Chapter 6] a technique to show that two events are correlated, is described.

The following lemma gives a bound on the expected number of ending head jobs at time t.

Lemma 2. *At any time t, $\mathbf{E}[h(t)] \le K - k + 2$.*

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

We next consider the expected value of $h'(t)$. Let $H(t)$ denote the ordered sequence $(q_0, \dots, q_{k-1})$ of jobs that are at time t at the head of the first k queues $Q_0, \dots, Q_{k-1}$, respectively. We use $q_i = \times$ to denote that Q_i is empty at time t. We define a binary variable $H_i(t)$ as follows: $H_i(t) = 1$ if $q_i \neq \times$ and q_i is in its final queue; $H_i(t) = 0$ otherwise. Let $H \in (J \cup \times)^k$ denote any possible configuration for $H(t)$. Observe that by definition $\mathbf{P}[H_i(t) = 1 \,|\, H(t) = H] = 0$ if $q_i = \times$. Let $q_i \neq \times$, then

$$\mathbf{P}[H_i(t) = 1 \,|\, H(t) = H] = \mathbf{P}[p_{q_i} \le 2^i \,|\, H(t) = H].$$

Since the two events $(p_{q_i} \le 2^i)$ and $(H(t) = H)$ are negatively correlated, we have that $\mathbf{P}[p_{q_i} \le 2^i \,|\, H(t) = H] \le \mathbf{P}[p_{q_i} \le 2^i]$.

Now, if a job q_i is of class larger than k we have $\mathbf{P}[p_{q_i} \le 2^i] = 0$. Otherwise, since the underlying probability distribution is well-shaped, we have (i) $\mathbf{P}[p_{q_{k-1}} \le 2^{k-1}] < 1/2$, and (ii) $\mathbf{P}[p_{q_i} \le 2^i] \le \frac{1}{2}\mathbf{P}[p_{q_{i+1}} \le 2^{i+1}]$, for all $0 \le i < k-1$. As a consequence, we obtain $\mathbf{P}[p_{q_i} \le 2^i] < \frac{1}{2^{k-i}}$ for all $0 \le i \le k-1$. Thus,

$$\mathbf{E}[h'(t) \,|\, H(t) = H] = \sum_{i=0}^{k-1} \mathbf{P}[H_i(t) = 1 \,|\, H(t) = H] < \frac{1}{2^k}\sum_{i=0}^{k-1} 2^i = \frac{2^k - 1}{2^k} < 1.$$

And therefore,

$$\mathbf{E}[h'(t)] = \sum_{H \in (J\cup\times)^k} \mathbf{E}[h'(t) \,|\, H(t) = H]\,\mathbf{P}[H(t) = H] < \sum_{H \in (J\cup\times)^k} \mathbf{P}[H(t) = H] = 1.$$

□

We also need the following bound on the probability that the sum of the random parts of the processing times exceeds a certain threshold value.

Lemma 3. $\mathbf{P}\left[\sum_{j\in J} \epsilon_j \ge \frac{n(2^k+1)}{8}\right] \ge 1 - e^{-\frac{n}{16}}$.

Proof (sketch). The lemma follows from applying a Chernoff bound on $\sum_j (\epsilon_j \ge 2^{k-1} + \frac{1}{2})$. □

We are now ready to prove Theorem 1. For the sake of conciseness, we introduce the following notation. Let $\alpha = (\sigma/2^k)^2$. For an instance I, we define $\mathcal{D}_I = \{t : \delta_I^{\mathcal{A}}(t) \leq \frac{2}{\alpha}\delta_I^l(t)\}$ and $\bar{\mathcal{D}}_I = \{t : \delta_I^{\mathcal{A}}(t) > \frac{2}{\alpha}\delta_I^l(t)\}$. Moreover, we define the event

$$\mathcal{E} = \left(\sum_j p_j \geq \sum_j \phi_j + \tfrac{n(2^k+1)}{8}\right)$$

and use $\bar{\mathcal{E}}$ to refer to the complement of $\mathcal{E}$.

Theorem 1. *For any instance $\bar{I}$ and any well-shaped probability distribution function f,*

$$\mathbf{E}_{I \in_f N(\bar{I})}\left[\frac{F^{\mathcal{A}}}{F^{\mathcal{OPT}}}\right] = O\left(\left(\frac{2^k}{\sigma}\right)^3 + \left(\frac{2^k}{\sigma}\right)^2 2^{K-k}\right).$$

Proof. In the following we omit that the expectation is taken over a distribution f in $N(\bar{I})$.

$$\begin{aligned}\mathbf{E}\left[\frac{F^{\mathcal{A}}}{F^{\mathcal{OPT}}}\right] &= \mathbf{E}\left[\frac{F^{\mathcal{A}}}{F^{\mathcal{OPT}}}\,\middle|\,\mathcal{E}\right]\mathbf{P}[\mathcal{E}] + \mathbf{E}\left[\frac{F^{\mathcal{A}}}{F^{\mathcal{OPT}}}\,\middle|\,\bar{\mathcal{E}}\right]\mathbf{P}[\bar{\mathcal{E}}] \\ &\leq \mathbf{E}\left[\frac{F^{\mathcal{A}}}{F^{\mathcal{OPT}}}\,\middle|\,\mathcal{E}\right]\mathbf{P}[\mathcal{E}] + ne^{-\frac{n}{16}},\end{aligned}$$

where the inequality follows from Lemma 3. Let $c = \frac{16}{e}$, then $ne^{-\frac{n}{16}} \leq c$. We partition the flow time $F^{\mathcal{A}} = \int_t \delta^{\mathcal{A}}(t)dt$ into the contribution of time instants $t \in \mathcal{D}$ and $t \in \bar{\mathcal{D}}$, i.e., $F^{\mathcal{A}} = \int_{t\in\mathcal{D}} \delta^{\mathcal{A}}(t)dt + \int_{t\in\bar{\mathcal{D}}} \delta^{\mathcal{A}}(t)dt$, and bound these contributions separately.

$$\begin{aligned}\mathbf{E}\left[\frac{\int_{t\in\mathcal{D}}\delta^{\mathcal{A}}(t)dt}{F^{\mathcal{OPT}}}\,\middle|\,\mathcal{E}\right]\mathbf{P}[\mathcal{E}] &\leq \mathbf{E}\left[\frac{\int_{t\in\mathcal{D}}\frac{2}{\alpha}\delta^l(t)dt}{F^{\mathcal{OPT}}}\,\middle|\,\mathcal{E}\right]\mathbf{P}[\mathcal{E}] \\ &\leq \mathbf{E}\left[\frac{\int_{t\in\mathcal{D}}\frac{2}{\alpha}h(t)dt + \int_{t\in\mathcal{D}}\frac{2}{\alpha}\cdot 6\beta_k\delta^{\mathcal{OPT}}(t)dt}{F^{\mathcal{OPT}}}\,\middle|\,\mathcal{E}\right]\mathbf{P}[\mathcal{E}] \\ &\leq \mathbf{E}\left[\frac{\int_{t\in\mathcal{D}}\frac{2}{\alpha}h(t)dt}{F^{\mathcal{OPT}}}\,\middle|\,\mathcal{E}\right]\mathbf{P}[\mathcal{E}] + \frac{2}{\alpha}\cdot 6\beta_k,\end{aligned}$$

where we use the deterministic bound of Lemma 1 on $\delta^l(t)$ and the fact that $F^{\mathcal{OPT}} \geq \int_{t\in\mathcal{D}}\delta^{\mathcal{OPT}}(t)dt$. We continue by exploiting the fact that given $\mathcal{E}$, $F^{\mathcal{OPT}} \geq \sum_j p_j \geq \sum_j \phi_j + \frac{n(2^k+1)}{8}$.

$$\begin{aligned}\mathbf{E}\left[\frac{\int_{t\in\mathcal{D}}\delta^{\mathcal{A}}(t)dt}{F^{\mathcal{OPT}}}\,\middle|\,\mathcal{E}\right]\mathbf{P}[\mathcal{E}] &\leq \frac{\mathbf{E}\left[\int_{t\in\mathcal{D}}\frac{2}{\alpha}h(t)dt\,|\,\mathcal{E}\right]\mathbf{P}[\mathcal{E}]}{\sum_j\phi_j + \frac{n(2^k+1)}{8}} + \frac{2}{\alpha}\cdot 6\beta_k \\ &\leq \frac{\frac{2}{\alpha}(K-k+2)\mathbf{E}[\sum_j p_j]}{\sum_j\phi_j + \frac{n(2^k+1)}{8}} + \frac{2}{\alpha}\cdot 6\beta_k,\end{aligned}$$

where we use Lemma 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 $\mathbf{E}[\sum_j p_j] = \sum_j \phi_j + \frac{n(2^k+1)}{2}$, we obtain,

$$\mathbf{E}\left[\frac{\int_{t\in\mathcal{D}}\delta^{\mathcal{A}}(t)dt}{F^{\mathcal{OPT}}}\,\middle|\,\mathcal{E}\right]\mathbf{P}[\mathcal{E}] \leq \frac{2}{\alpha}\cdot 4(K-k+2) + \frac{2}{\alpha}\cdot 6\beta_k.$$

For $t \in \bar{\mathcal{D}}$, by the fact that given $\mathcal{E}$, $F^{\mathcal{OPT}} \geq \sum_j \phi_j + \frac{n(2^k+1)}{8}$, and by exploiting Lemma 4, which is given below, we obtain

$$\mathbf{E}\left[\frac{\int_{t\in\bar{\mathcal{D}}} \delta^{\mathcal{A}}(t)dt}{F^{\mathcal{OPT}}}\,\middle|\,\mathcal{E}\right]\mathbf{P}[\mathcal{E}] \leq \frac{\mathbf{E}[\int_{t\in\bar{\mathcal{D}}} \delta^{\mathcal{A}}(t)dt\,|\,\mathcal{E}]\mathbf{P}[\mathcal{E}]}{\sum_j \phi_j + \frac{n(2^k+1)}{8}} \leq \frac{\frac{8}{\alpha}\mathbf{E}\left[\sum_j p_j\right]}{\sum_j \phi_j + \frac{n(2^k+1)}{8}} \leq \frac{32}{\alpha}.$$

Putting everything together, we obtain

$$\mathbf{E}\left[\frac{F^{\mathcal{A}}}{F^{\mathcal{OPT}}}\right] \leq \frac{2}{\alpha}\cdot 4(K-k+2) + \frac{2}{\alpha}\cdot 6\beta_k + \frac{32}{\alpha} + c = O\left(\left(\frac{2^k}{\sigma}\right)^3 + \left(\frac{2^k}{\sigma}\right)^2 2^{K-k}\right),$$

where the last equality follows from the definition of α and β_k.

□

To finalize the proof we are left to show that the following lemma holds.

Lemma 4. $\mathbf{E}[\int_{t\in\bar{\mathcal{D}}} \delta^{\mathcal{A}}(t)dt\,|\,\mathcal{E}]\,\mathbf{P}[\mathcal{E}] \leq \frac{8}{\alpha}\mathbf{E}\left[\sum_j p_j\right]$.

4.3 Proof of Lemma 4

We only provide an overview of the proof of Lemma 4 here. The complete proof requires a number of additional techniques and lemmas that are provided in the appendix.

The following two lemmas bound the probability that a job is lucky. In the first one, we prove that a job j with $\bar{p}_j \in (2^{i-1}, 2^i]$, for some $i > k$, is lucky with probability at least $\frac{1}{2}$.

Lemma 5. *For each job j with $\bar{p}_j \in (2^{i-1}, 2^i]$, for some i, $k < i \leq K$,* $\mathbf{P}[X_j^l = 1] \geq \frac{1}{2}$.

Proof (sketch). Follows directly from the definition of well-shaped distributions. □

We now show that the probability of a job j being lucky given that it is of class i, $i \leq k$, is at least $\alpha = (\sigma/2^k)^2$.

Lemma 6. *For each job j with $\bar{p}_j \in [1, 2^k]$ and each class i, $0 \leq i \leq k$,* $\mathbf{P}[X_j^l = 1\,|\,Cl_j = i] \geq \alpha$.

Proof (sketch). The only difficult part is for $Cl_j = k$. For $\gamma_k \leq \frac{1}{\sqrt{2}}\left(\frac{\sigma}{2^{k-1}}\right)$, we can show an "Inverse Chebyshev" inequality, from which the lemma follows. □

It is easy to see that Lemma 6 can be tightened so that we achieve probability at least $\frac{1}{2}$ on the uniform distribution.

In the rest of this section we only consider properties of the schedule $\mathcal{A}$ produced by MLF. We therefore omit the superscript $\mathcal{A}$ in the notation below.

Let $S \subseteq J$. In the following, we will condition on the event that (i) the set of active jobs at time t is equal to S, i.e., $(S(t) = S)$, and (ii) the processing times of all jobs not in S are fixed to values that are described by a vector $\mathbf{x}_{\bar{S}}$, which we denote by $(\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}})$. For the sake of conciseness, we define the event $\mathcal{F}(t, S, \mathbf{x}_{\bar{S}}) = ((S(t) = S) \cap (\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}}))$. Observe that $\mathbf{P}[X_j^l(t) = 1\,|\,\mathcal{F}(t, S, \mathbf{x}_{\bar{S}})] = 0$ if $j \notin S$, since j is not alive at time t. Moreover, $\mathbf{P}[X_j^l(t) = 1\,|\,\mathcal{F}(t, S, \mathbf{x}_{\bar{S}})] = \mathbf{P}[X_j^l = 1\,|\,\mathcal{F}(t, S, \mathbf{x}_{\bar{S}})]$ if $j \in S$. Thus,

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

Conditioned on $\mathcal{F}(t, S, \mathbf{x}_{\bar{S}})$, we first show that the expected number of jobs that are lucky and alive at time t is at least a good fraction of the number of jobs that are alive at that time.

Lemma 7. *For every $j \in S$, $\mathbf{P}[X_j^l = 1 \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}})] \geq \alpha$. Therefore, $\mathbf{E}[\delta^l(t) \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}})] \geq \alpha|S|$.*

Proof. Let $\bar{p}_j \in (2^{i-1}, 2^i]$, for some i, $k < i \leq K$. The events $(X_j^l = 1)$ and $(\mathcal{F}(t, S, \mathbf{x}_{\bar{S}}))$ are positively correlated and thus,

$$\mathbf{P}[X_j^l = 1 \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}})] \geq \mathbf{P}[X_j^l = 1] .$$

Next, let $\bar{p}_j \in [1, 2^k]$. The events $(X_j^l = 1 \,|\, Cl_j = i)$ and $(\mathcal{F}(t, S, \mathbf{x}_{\bar{S}}) \,|\, Cl_j = i)$ are positively correlated for all i, $0 \leq i \leq k$, i.e.,

$$\mathbf{P}[X_j^l = 1 \cap \mathcal{F}(t, S, \mathbf{x}_{\bar{S}}) \,|\, Cl_j = i] \geq \mathbf{P}[X_j^l = 1 \,|\, Cl_j = i] \, \mathbf{P}[\mathcal{F}(t, S, \mathbf{x}_{\bar{S}}) \,|\, Cl_j = i] .$$

Thus,

$$\begin{aligned}
\mathbf{P}[X_j^l = 1 \cap \mathcal{F}(t, S, \mathbf{x}_{\bar{S}})] &= \sum_{i=0}^{k} \mathbf{P}[X_j^l = 1 \cap \mathcal{F}(t, S, \mathbf{x}_{\bar{S}}) \,|\, Cl_j = i] \, \mathbf{P}[Cl_j = i] \\
&\geq \sum_{i=0}^{k} \mathbf{P}[X_j^l = 1 \,|\, Cl_j = i] \, \mathbf{P}[\mathcal{F}(t, S, \mathbf{x}_{\bar{S}}) \,|\, Cl_j = i] \, \mathbf{P}[Cl_j = i] \\
&\geq \min_{i=0,\ldots,k} \mathbf{P}[X_j^l = 1 \,|\, Cl_j = i] \, \mathbf{P}[\mathcal{F}(t, S, \mathbf{x}_{\bar{S}})] .
\end{aligned}$$

And therefore,

$$\mathbf{P}[X_j^l = 1 \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}})] \geq \min_{i=0,\ldots,k} \mathbf{P}[X_j^l = 1 \,|\, Cl_j = i] .$$

The lemma follows from Lemmas 5 and 6.

□

We use the previous lemma to prove that, with high probability, at any time t the number of lucky jobs is also a good fraction of the overall number of jobs in the system.

Lemma 8. *For any $S \subseteq J$, at any time t, $\mathbf{P}[\delta^l(t) < \frac{1}{2}\alpha\delta(t) \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}})] \leq e^{-\frac{\alpha|S|}{8}}$.*

Proof (sketch). Given $\mathcal{F}(t, S, \mathbf{x}_{\bar{S}})$, we will first show that the variables $(X_j^l \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}}))$, $j \in S$, are independent. The proof follows by applying a Chernoff bound to $\sum_{j \in S}(X_j^l \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}}))$, and by using Lemma 7 to bound the expected value of the sum. □

Corollary 1. *For any $s = 1, \ldots, n$, at any time t, $\mathbf{P}[\delta^l(t) < \frac{1}{2}\alpha\delta(t) \,|\, \delta(t) = s] \leq e^{-\frac{\alpha s}{8}}$.*

We are now ready to prove Lemma 4.

Proof.

$$\begin{aligned}
\mathbf{E}\left[\int_{t \in \bar{\mathcal{D}}} \delta^{\mathcal{A}}(t) dt \,\middle|\, \mathcal{E}\right] \mathbf{P}[\mathcal{E}] &\leq \mathbf{E}\left[\int_{t \in \bar{\mathcal{D}}} \delta^{\mathcal{A}}(t) dt\right] = \int_{t \geq 0} \mathbf{E}\left[\delta^{\mathcal{A}}(t) \,|\, t \in \bar{\mathcal{D}}\right] \mathbf{P}[t \in \bar{\mathcal{D}}] \, dt \\
&= \int_{t \geq 0} \sum_{s=1}^{n} s \, \mathbf{P}[\delta^{\mathcal{A}}(t) = s \,|\, t \in \bar{\mathcal{D}}] \, \mathbf{P}[t \in \bar{\mathcal{D}}] \, dt
\end{aligned}$$

$$
\begin{aligned}
&= \int_{t\geq 0} \sum_{s=1}^{n} s\,\mathbf{P}[t \in \bar{\mathcal{D}} \,|\, \delta^{\mathcal{A}}(t) = s]\,\mathbf{P}[\delta^{\mathcal{A}}(t) = s]\,dt \\
&\leq \int_{t\geq 0} \sum_{s=1}^{n} s\,e^{-\frac{\alpha s}{8}}\,\mathbf{P}[\delta^{\mathcal{A}}(t) = s]\,dt \leq \frac{8}{\alpha}\int_{t\geq 0} \sum_{s=1}^{n} \mathbf{P}[\delta^{\mathcal{A}}(t) = s]\,dt \\
&= \frac{8}{\alpha}\int_{t\geq 0} \mathbf{P}[\delta^{\mathcal{A}}(t) \geq 1]\,dt = \frac{8}{\alpha}\mathbf{E}[\textstyle\sum_j p_j],
\end{aligned}
$$

where the fifth inequality is due to Corollary 1 and the sixth inequality follows since $e^{-x} < \frac{1}{x}$, for $x > 0$.

□

4.4 Adaptive Adversary

Recall that an adaptive adversary may change its input instance on basis of the outcome of the random process. Lemmas 2 and 7 are those in which an adaptive adversary might change the analysis with respect to an oblivious one. In Appendix C we discuss why these lemmas also hold for an adaptive adversary. Thus, the upper bound on the smoothed competitive ratio given in Theorem 1 also holds against an adaptive adversary.

5 Lower Bounds

The first bound is an $\Omega(2^{K/6-k/2})$ one on the smoothed competitive ratio for any deterministic algorithm against an oblivious adversary.

Theorem 2. *Any deterministic algorithm $\mathcal{A}$ 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.*

As mentioned in the introduction, 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. The next theorem gives an $\Omega(2^{K-k})$ lower bound on the smoothed competitive ratio of any deterministic algorithm 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. in [17] for an $\Omega(2^K)$ non-clairvoyant deterministic lower bound.

Theorem 3. *Any deterministic algorithm $\mathcal{A}$ has smoothed competitive ratio $\Omega(2^{K-k})$ against an adaptive adversary in the partial bit randomization smoothening model.*

For other smoothening models, we only provide lower bounds on the performance of MLF. The models, as defined in Section 2, can all be captured using the symmetric smoothening model according to φ. Consider a function $\varphi : \mathbb{R}^+ \to \mathbb{R}^+$, which is continuous and non-decreasing. The symmetric smoothening model according to φ smoothens the original processing times as follows: $p_j = \max(1, \bar{p}_j + \epsilon_j)$, where ϵ_j is chosen randomly from $[-\varphi(\bar{p}_j)/2, \varphi(\bar{p}_j)/2]$ according to the uniform probability distribution f.

Theorem 4. *Let $\varphi : \mathbb{R}^+ \to \mathbb{R}^+$ be function such that $\varphi(y) < 2^{K-2}$ for all y, and let $a \geq 1$ such that there exist $x \in \mathbb{R}^+$ satisfying $x + \varphi(x)/2 = 2^{K-1} + a$. Then, there exists an $\Omega(2^K/a)$ lower bound on the smoothed competitive ratio of MLF against an oblivious adversary in the symmetric smoothening model according to φ.*

The additive symmetric smoothening model is equivalent to the above defined model with $\varphi(y) = c$, for $c \leq 2^{K-2}$. If ϵ_j is drawn using a uniform distribution, we can set $a = 1$ and $x = 2^{K-1} + 1 - c/2$. This way, we obtain an $\Omega(2^K)$ lower bound for this model against an oblivious adversary.

For the additive relative symmetric smoothening model, we define $\varphi(x) = x^c$, for $c \leq \frac{K-2}{\log(3 \cdot 2^{K-3}+1)}$. Choosing x such that $x + \frac{1}{2}x^c = 2^{K-1} + 1$ and $a = 1$ and drawing ϵ_j from the uniform distribution, we have an $\Omega(2^K)$ lower bound for this model.

For the multiplicative model, we define $\varphi(x) = \epsilon x$, for $\epsilon \in [0, \frac{2^{K-2}}{3 \cdot 2^{K-3}+1}]$. Drawing ϵ_j from the uniform distribution, we have for $a = 1$, $x = (2^K + 2)/(2 + \epsilon)$. Thus, there is an $\Omega(2^K)$ lower bound for this smoothening model.

Obviously, Theorem 4 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 Concluding Remarks

In this paper, we analyzed the performance of the Multi-Level 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 smoothening models, including the additive symmetric one, which adapts to our case the model introduced by Spielman and Teng [23]. 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 probability density function for the random perturbation. The analysis holds for any well-shaped probability distribution. 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 must have a smoothed competitive ratio of $\Omega(2^{K-k})$. Hence, MLF is optimal up to a constant factor in this model. For the other proposed smoothening 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 consider a weaker definition of smoothed competitiveness as the ratio between the expected costs of the algorithm and of the optimum, see also [21], rather than the expected competitive ratio. We remark that from Lemmas 1, 2, 5 and 6 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 smoothened, 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 1 having an extra factor of K, which will also be in the smoothed competitive ratio. Finally, this framework of analysis could be extended to other online problems.

Acknowledgements

The authors would like to thank Alessandro Panconesi, Kirk Pruhs and Gerhard Woeginger for helpful discussions and suggestions.

References

[1] N. Alon and J. H. Spencer. *The Probabilistic Method.* Wiley-Interscience Series in Discrete Mathematics. Wiley, Chichester, second edition, 2000.

[2] C. Banderier, R. Beier, and K. Mehlhorn. Smoothed analysis of three combinatorial problems. In *28th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, 2003.

[3] L. Becchetti and S. Leonardi. Non-clairvoyant scheduling to minimize the average flow time on single and parallel machines. In *Proceedings of the Thirty-Third Annual ACM Symposium on the Theory of Computing*, pages 94–103, 2001.

[4] Luca Becchetti, Stefano Leonardi, Alberto Marchetti-Spaccamela, Guido Schäfer, and Tjark Vredeveld. Average case and smoothed competitive analysis of the multi-level feedback algorithm. Technical Report MPI–I–2003–1–014, Max-Planck-Institut für Informatik, Saarbrücken, Germany, 2003.

[5] R. Beier, P. Krysta, and B. Vöcking. Computing equilibria for congestion games with (im)perfect information. Unpublished manuscript, 2003.

[6] A. Blum and J. Dunagan. Smoothed analysis of the perceptron algorithm. In *In Proceedings of 13th annual ACM-SIAM Symposium on Discrete Algorithms*, pages 905–914, 2002.

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

[8] A. Borodin, S. Irani, P. Raghavan, and B. Schieber. Competitive paging with locality of reference. *Journal of Computer and System Sciences*, 50(2):244–258, 1995.

[9] D. Dubhashi and A. Panconesi. Concentration of measure for the analysis of randomised algorithms. Unpublished manuscript, `http://www.dsi.uniroma1.it/~ale/Papers/master.ps`, 2003.

[10] J. Dunagan, D. A. Spielman, and S. H. Teng. Smoothed analysis of the Renegar's condition number for linear programming. `http://www-math.mit.edu/~spielman/SmoothedAnalysis`, accessed 2002.

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

[12] B. Kalyanasundaram and K. Pruhs. Minimizing flow time nonclairvoyantly. In *In Proceedings of the Thirty-Eight IEEE Symposium on Foundations of Computer Science*, pages 345–352, 1997. To appear in Journal of the ACM.

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

[14] E. Koutsoupias and C. Papadimitriou. Beyond competitive analysis. In *Proceedings of the Twenty-Fifth Symposium on Foundations of Computer Science*, pages 394–400, 1994.

[15] S. Leonardi and D. Raz. Approximating total flow time on parallel machines. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 110–119, 1997.

[16] J. E. Michel and E. G. Coffman. Synthesis of a feedback queueing discipline for computer operation. *Journal of the ACM*, 21:329–339, 1974.

[17] R. Motwani, S. Phillips, and E. Torng. Non-clairvoyant scheduling. *Theoretical Computer Science*, 130:17–47, 1994.

[18] R. Motwani and P. Raghavan. *Randomized algorithms.* Cambridge University Press, Cambridge, first edition, 1995.

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

[20] A. Sankar, D. A. Spielman, and S. H. Teng. Smoothed analysis of the condition numbers and growth factors of matrices. `http://www-math.mit.edu/~spielman/SmoothedAnalysis`, accessed 2002.

[21] M. Scharbrodt, T. Schickinger, and A. Steger. A new average case analysis for completion time scheduling. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 170–178, 2002.

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

[23] D. Spielman and S. H. Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, pages 296–305, 2001.

[24] D. Spielman and S. H. Teng. Smoothed analysis of property testing. `http://www-math.mit.edu/~spielman/SmoothedAnalysis`, 2002.

[25] D. Spielman and S. H. Teng. Smoothed analysis of interior-point algorithms: Termination. `http://www-math.mit.edu/~spielman/SmoothedAnalysis`, submitted, 2003.

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

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., [11].

Theorem 5. *Let $X_1, \dots, X_n$ 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 6. *Let X be the sum of a finite number of mutually independent binary random variables such that $\mu = \mathbf{E}[X]$ is positive. Then,*

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

Theorem 7. *Let X be the sum of a finite number of mutually independent binary random variables such that $\mu = \mathbf{E}[X]$ is positive. Then,*

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

Theorem 8. *Let X be the sum of a finite number of mutually independent binary random variables such that $\mu = \mathbf{E}[X]$ is positive. Then,*

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

B Proof of Lemma 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, that is the overall time the machine has been processing jobs until time t. For a generic function f (δ, V or L), we define $\Delta f(t) = f^{\mathcal{A}}(t) - f^{\mathcal{OPT}}(t)$. For f (δ, V, ΔV, L or ΔL), the notation $f_{=k}(t)$ will denote the value of function f at time t when restricted to jobs of class exactly k. We use $f_{\ge h, \le k}(t)$ to denote the value of f at time t when restricted to jobs of classes between h and k.

Lemma 1. *For any input instance I, at any time t: $\delta_I^l(t) \le h_I(t) + 6\beta_k \delta_I^{\mathcal{OPT}}(t)$.*

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. For $\delta^l(t)$, we write the following relations:

$$\delta^l(t) \le h(t) + \beta_k \sum_{i=k_1}^{k_2} \frac{V_{=i}^{\mathcal{A}}(t)}{2^{i-1}}. \tag{1}$$

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 we can therefore not say 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 $2^{i-1}/\beta_k$. We continue with:

$$\begin{aligned}
\sum_{i=k_1}^{k_2} \frac{V^{\mathcal{A}}_{=i}(t)}{2^{i-1}} &= \sum_{i=k_1}^{k_2} \frac{V^{\mathcal{OPT}}_{=i}(t) + \Delta V_{=i}(t)}{2^{i-1}} \\
&\leq 2\delta^{\mathcal{OPT}}_{\geq k_1, \leq k_2}(t) + \sum_{i=k_1}^{k_2} \frac{\Delta V_{=i}(t)}{2^{i-1}} \\
&= 2\delta^{\mathcal{OPT}}_{\geq k_1, \leq k_2}(t) + 2\sum_{i=k_1}^{k_2} \frac{\Delta V_{\leq i}(t) - \Delta V_{\leq i-1}(t)}{2^{i}} \\
&= 2\delta^{\mathcal{OPT}}_{\geq k_1, \leq k_2}(t) + 2\frac{\Delta V_{\leq k_2}(t)}{2^{k_2}} + 2\sum_{i=k_1}^{k_2-1} \frac{\Delta V_{\leq i}(t)}{2^{i+1}} \\
&\leq 2\delta^{\mathcal{OPT}}_{\geq k_1, \leq k_2}(t) + \delta^{\mathcal{OPT}}_{\leq k_1-1}(t) + 4\sum_{i=k_1}^{k_2} \frac{\Delta V_{\leq i}(t)}{2^{i+1}} \\
&\leq 2\delta^{\mathcal{OPT}}_{\leq k_2}(t) + 4\sum_{i=k_1}^{k_2} \frac{\Delta V_{\leq i}(t)}{2^{i+1}}, \qquad (2)
\end{aligned}$$

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

We are left to study the sum in (2). For any $t_1 \leq t_2 \leq t$, for a generic function f, denote by $f^{[t_1,t_2]}(t)$ the value of function f at time t when restricted to jobs released between t_1 and t_2, e.g., $L^{[t_1,t_2]}_{\leq i}(t)$ 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 $i = k_1, \ldots, k_2$, $[t_{i+1}, t) \supseteq [t_i, t)$.

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 $Q_0, \ldots, Q_i$. Therefore,

$$\Delta V_{\leq i}(t) \leq \Delta V^{(t_i,t]}_{\leq i}(t) \leq L^{\mathcal{A}(t_i,t]}_{>i}(t) - L^{\mathcal{OPT}(t_i,t]}_{>i}(t) = \Delta L^{(t_i,t]}_{>i}(t).$$

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

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

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 $(t_{j+1}, t_j]$, $j = k_1 - 1, \ldots, i-1$.

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

$$
\begin{aligned}
\sum_{i=k_1}^{k_2} \frac{\Delta L_{>i}^{(t_i,t]}(t)}{2^{i+1}} &\leq \sum_{j=k_1-1}^{k_2-1} \sum_{i=j+1}^{k_2} \frac{L_{>\bar{i}(j)}^{\mathcal{A}(t_{j+1},t_j]}(t) - L_{>\bar{i}(j)}^{\mathcal{OPT}(t_{j+1},t_j]}(t)}{2^{i+1}} \\
&\leq \sum_{j=k_1-1}^{k_2-1} \frac{L_{>\bar{i}(j)}^{\mathcal{A}(t_{j+1},t_j]}(t) - L_{>\bar{i}(j)}^{\mathcal{OPT}(t_{j+1},t_j]}(t)}{2^{j+1}} \\
&\leq \sum_{j=k_1-1}^{k_2-1} \delta_{>\bar{i}(j)}^{\mathcal{OPT}(t_{j+1},t_j]}(t) \;\leq\; \delta_{\geq k_1}^{\mathcal{OPT}(t_{k_2},t]}(t) \\
&\leq \delta_{\geq k_1}^{\mathcal{OPT}}(t). \qquad (3)
\end{aligned}
$$

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 $x_j^{\mathcal{A}}(t)$. Now, observe that each of these jobs has initial processing time at least $2^{\bar{i}(j)} \geq 2^{j+1}$ at their release and we give to the optimum the further advantage that it finishes every such job when processed for an amount $x_j^{\mathcal{A}}(t) \leq 2^{j+1}$. To maximize the number of finished jobs the optimum places the work $L_{>\bar{i}(j)}^{\mathcal{OPT}(t_{j+1},t_j]}$ on the jobs with smaller $x_j^{\mathcal{A}}(t)$. The optimum is then left at time t with a number of jobs

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

Altogether, from (1), (2) and (3) we obtain:

$$
\begin{aligned}
\delta^l(t) &\leq h(t) + 2\beta_k \delta_{\leq k_2}^{\mathcal{OPT}}(t) + 4\beta_k \delta_{\geq k_1}^{\mathcal{OPT}}(t) \\
&\leq h(t) + 6\beta_k \delta^{\mathcal{OPT}}(t).
\end{aligned}
$$

□

C Lattice Argument

In the sequel, we exploit the fact that two events A and B are correlated: A and B are *positively correlated* if $\mathbf{P}[A \cap B] \geq \mathbf{P}[A]\,\mathbf{P}[B]$, while A and B are *negatively correlated* if $\mathbf{P}[A \cap B] \leq \mathbf{P}[A]\,\mathbf{P}[B]$. We briefly review a technique described in the book by Alon and Spencer [1, Chapter 6] to show that two events are correlated. Then, we discuss the use of this technique in our analysis.

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 $(\Omega, \leq, \vee, \wedge)$ is a partially ordered set $(\Omega, \leq)$ 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 $(\Omega, \leq, \vee, \wedge)$ 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 $x, y \in \Omega$,

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

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

Example 1 *(Lemma 7).* Let $A' = (X_j^l = 1 \,|\, Cl_j = i)$ and $B' = (\mathcal{F}(t, S, \mathbf{x}_{\bar{S}}) \,|\, Cl_j = i)$. We condition the probability space further in order to make sure that only the processing time of j is random. That is, we fix the processing times of all jobs different from j to $\mathbf{x}_{\bar{j}}$, which we denote by $\mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}}$. Define $A = (A' \,|\, \mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}}) = (X_j^l = 1 \,|\, Cl_j = i \cap \mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}})$ and $B = (B' \,|\, \mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}}) = (\mathcal{F}(t, S, \mathbf{x}_{\bar{S}}) \,|\, Cl_j = i \cap \mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}})$. Let Ω denote the (conditioned) probability space and let $\mathbf{P}$ denote the underlying (conditioned) probability distribution.

(i) It is easy to see that Ω together with the partial order $\leq$, 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.

(iii) Let $p_j = x$ and assume $x \in A$, i.e., j is lucky with respect to $p_j = x$. If we increase p_j to $y \geq x$, then j will remain lucky and thus $y \in A$. So, A is monotone increasing. Similarly, if $\mathcal{F}(t, S, \mathbf{x}_{\bar{S}})$ holds for $p_j = x$, then $\mathcal{F}(t, S, \mathbf{x}_{\bar{S}})$ holds also for $p_j = y \geq x$, since the two schedules obtained are the same up to time t. That is, B is monotone increasing. We conclude that A and B are positively correlated.

Note that A' and $(\mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}})$ are mutually independent and thus $\mathbf{P}[A'] = \mathbf{P}[A]$. We exploit this fact as follows in order to prove that the events A' and B' are positively correlated as well.

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

Example 2 *(Lemma 2).* If we define $A' = (p_{q_i} \leq 2^i)$ and $B' = (H(t) = H)$ and then proceed along the lines of Example 1 it is easy to see that A is monotone decreasing and B is monotone increasing. That is, A and B are negatively correlated. Negative correlation of A' and B' then follows from the observation that A' is independent of $(\mathbf{p}_{\bar{j}} = \mathbf{x}_{\bar{j}})$.

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 behaviour of the adaptive adversary will be the same.

D Proof of Lemma 3

Lemma 3. $\mathbf{P}\left[\sum_{j\in J}\epsilon_j \geq \frac{n(2^k+1)}{8}\right] \geq 1-e^{-\frac{n}{16}}$.

Proof. For each job $j \in J$, we define a binary random variable $Z_j = (\epsilon_j \geq \frac{2^k+1}{2})$. Let $Z = \sum_{j\in J} Z_j$. We have $\mathbf{P}[Z_j = 1] = \frac{1}{2}$ and therefore $\mathbf{E}[Z] = \frac{n}{2}$. Since each ϵ_j is chosen independently uniformly at random, the Z_j's are independent. Applying a Chernoff bound we obtain $\mathbf{P}[Z < \frac{n}{4}] \leq e^{-\frac{n}{16}}$. Thus, with probability at least $1-e^{-\frac{n}{16}}$ there are at least $\frac{n}{4}$ jobs with $\epsilon_j \geq \frac{2^k+1}{2}$. □

E Proof of Lemma 5

Lemma 5. *For each job j with $\bar{p}_j \in (2^{i-1}, 2^i]$, for some i, $k < i \leq K$, $\mathbf{P}[X_j^l = 1] \geq \frac{1}{2}$.*

Proof. Due to Fact 2 the processing time p_j of a job j is chosen randomly from a subrange of $(2^{i-1}, 2^i]$. Hence,

$$\mathbf{P}[X_j^l = 1] = \mathbf{P}[p_j \geq (1+\gamma_k)2^{i-1}] \geq \mathbf{P}[\epsilon_j \geq \gamma_k 2^{i-1}],$$

since the worst case occurs if $\phi_j = 2^{i-1}$. By definition $\gamma_k \leq 2^{k-K}$ and thus $\gamma_k 2^{i-1} \leq \mu$ for each i. Since the underlying probability distribution is symmetric around its mean, j is lucky with probability at least $\frac{1}{2}$. □

F Proof of Lemma 6

Lemma 6. *For each job j with $\bar{p}_j \in [1, 2^k]$ and each class i, $0 \leq i \leq k$,*

$$\mathbf{P}[X_j^l = 1 \,|\, Cl_j = i] \geq \left(\frac{\sigma}{2^k}\right)^2.$$

Proof. Due to Fact 1 the processing time p_j of a job j is chosen completely at random from $[1, 2^k]$. Thus, $\mathbf{P}[X_j^l = 1 \,|\, Cl_j = i] = \mathbf{P}[\epsilon_j \geq (1+\gamma_k)2^{i-1} \,|\, Cl_j = i]$.

First, note that for each $i < k$, $\mathbf{P}[X_j^l = 1 \,|\, Cl_j = i] \geq \frac{1}{2}$, since $\gamma_k \leq \frac{1}{2}$ and the probability density function f is non-decreasing in $[1, \mu]$.

Next, let $i = k$. Then, $\mathbf{P}[\epsilon_j \geq (1+\gamma_k)2^{k-1} \,|\, Cl_j = k] \geq \mathbf{P}[\epsilon_j \geq (1+\gamma_k)2^{k-1}]$, since $\mathbf{P}[Cl_j = k] \leq 1$. Moreover, note that

$$\mathbf{P}[\epsilon_j \geq (1+\gamma_k)2^{k-1}] > \mathbf{P}[\epsilon_j - \mu \geq \gamma_k 2^{k-1}] = \tfrac{1}{2}\mathbf{P}[|\epsilon_j - \mu| \geq \gamma_k 2^{k-1}],$$

where the first inequality holds since we assume that $\mu > 2^{k-1}$ and the last inequality is due to the symmetry of the distribution. The lemma now follows from Corollary 2. □

The following lemma might be considered as the inverse of Chebyshev's inequality.

Lemma 9. *Let ϵ be drawn from a symmetric distribution over $[1, 2^k]$ with mean $\mu = 2^{k-1}+1/2$. Then, for any λ, $0 \leq \lambda \leq 2^k - \mu$,*

$$\mathbf{P}[|\epsilon - \mu| \geq \lambda] \geq \frac{\sigma^2 - \lambda^2}{(2^k-\mu)^2 - \lambda^2} \geq \left(\frac{\sigma}{2^{k-1}}\right)^2 - \left(\frac{\lambda}{2^{k-1}}\right)^2.$$

Proof.

$$\begin{aligned}\sigma^2 &= 2\int_{\mu}^{2^k}(\epsilon-\mu)^2 f(\epsilon)d\epsilon \;=\; 2\int_{\mu}^{\mu+\lambda}(\epsilon-\mu)^2 f(\epsilon)d\epsilon + 2\int_{\mu+\lambda}^{2^k}(\epsilon-\mu)^2 f(\epsilon)d\epsilon \\ &\leq \lambda^2(1-\mathbf{P}[|\epsilon-\mu|\geq\lambda]) + (2^k-\mu)^2\mathbf{P}[|\epsilon-\mu|\geq\lambda]\end{aligned}$$

□

Corollary 2. *For* $\gamma_k \leq \frac{1}{\sqrt{2}}\left(\frac{\sigma}{2^{k-1}}\right)$, *we have* $\mathbf{P}[|\epsilon_j-\mu|\geq\gamma_k 2^{k-1}] \geq \frac{1}{2}\left(\frac{\sigma}{2^{k-1}}\right)^2$.

G Proof of Lemma 7

Lemma 7. *For every* $j\in S$, $\mathbf{P}[X_j^l = 1\,|\,\mathcal{F}(t,S,\mathbf{x}_{\bar{S}})] \geq \alpha$. *Therefore,* $\mathbf{E}[\delta^l(t)\,|\,\mathcal{F}(t,S,\mathbf{x}_{\bar{S}})] \geq \alpha|S|$.

Proof. Let $\bar{p}_j \in (2^{i-1},2^i]$, for some i, $k<i\leq K$. The events $(X_j^l=1)$ and $(\mathcal{F}(t,S,\mathbf{x}_{\bar{S}}))$ are positively correlated and thus,

$$\mathbf{P}[X_j^l = 1\,|\,\mathcal{F}(t,S,\mathbf{x}_{\bar{S}})] \geq \mathbf{P}[X_j^l=1].$$

Next, let $\bar{p}_j\in[1,2^k]$. The events $(X_j^l=1\,|\,Cl_j=i)$ and $(\mathcal{F}(t,S,\mathbf{x}_{\bar{S}})\,|\,Cl_j=i)$ are positively correlated for all i, $0\leq i\leq k$, i.e.,

$$\mathbf{P}[X_j^l = 1\cap\mathcal{F}(t,S,\mathbf{x}_{\bar{S}})\,|\,Cl_j=i] \geq \mathbf{P}[X_j^l=1\,|\,Cl_j=i]\,\mathbf{P}[\mathcal{F}(t,S,\mathbf{x}_{\bar{S}})\,|\,Cl_j=i].$$

Thus,

$$\begin{aligned}\mathbf{P}[X_j^l = 1\cap\mathcal{F}(t,S,\mathbf{x}_{\bar{S}})] &= \sum_{i=0}^{k}\mathbf{P}[X_j^l=1\cap\mathcal{F}(t,S,\mathbf{x}_{\bar{S}})\,|\,Cl_j=i]\,\mathbf{P}[Cl_j=i] \\ &\geq \sum_{i=0}^{k}\mathbf{P}[X_j^l=1\,|\,Cl_j=i]\,\mathbf{P}[\mathcal{F}(t,S,\mathbf{x}_{\bar{S}})\,|\,Cl_j=i]\,\mathbf{P}[Cl_j=i] \\ &\geq \min_{i=0,\ldots,k}\mathbf{P}[X_j^l=1\,|\,Cl_j=i]\,\mathbf{P}[\mathcal{F}(t,S,\mathbf{x}_{\bar{S}})].\end{aligned}$$

And therefore,

$$\mathbf{P}[X_j^l=1\,|\,\mathcal{F}(t,S,\mathbf{x}_{\bar{S}})] \geq \min_{i=0,\ldots,k}\mathbf{P}[X_j^l=1\,|\,Cl_j=i].$$

The lemma follows from Lemma 5 and Lemma 6. □

H Proof of Lemma 8

We first prove that the variables $Y_j = (X_j^l\,|\,\mathcal{F}(t,S,\mathbf{x}_{\bar{S}}))$, $j\in S$, are independent.

Lemma 10. *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, given $\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}}$, there exist two possible, deterministic schedules $\mathcal{S}_1$ and $\mathcal{S}_2$, such that $S^{\mathcal{S}_1}(t) = S^{\mathcal{S}_2}(t) = S$. Denote by I_1 and I_2 the corresponding instances. Since the processing times of jobs not in S are fixed, it has to be the case that I_1 and I_2 differ in the processing times of some subset of the jobs in S. Let $t' \le t$ be the first time, where $\mathcal{S}_1$ and $\mathcal{S}_2$ differ. Since the job processed by MLF at time t' only depends on $S(t')$, it must be the case that $S_{I_1}(t') \neq S_{I_2}(t')$. This implies that one job j was completed right before t' in one schedule (we assume in $\mathcal{S}_1$ without loss of generality) but not in the other. Since j must belong to S and $t' \le t$, this contradicts the hypothesis that $S^{\mathcal{S}_1}(t) = S$. □

Corollary 3. *Assume $S(t) = S$ and $\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}}$. Then, for each $j \in S$, $x_j^{\mathcal{A}}(t)$ is a uniquely determined constant.*

In the sequel, given that $S(t) = S$ and $\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}}$, we set $\pi_j = x_j^{\mathcal{A}}(t)$ for all $j \in S$.

Fact 6. *Assume jobs in S are not completed before time t if MLF processes instance I. Then, for every instance I' that is obtained from I by increasing the processing times of some subset of the jobs in S, we have $x_{jI'}^{\mathcal{A}}(t) = x_{jI}^{\mathcal{A}}(t)$ for every job j.*

Lemma 11. *Assume $\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}}$ is such that the event $(S(t) = S)$ is non-empty. Then,*

$$(S(t) = S) \;\Leftrightarrow\; (p_j > \pi_j \textit{ for all } j \in S).$$

Proof. Assume $\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}}$ and consider any execution such that $(S(t) = S)$. By Lemma 10 and Corollary 3, we know that the amount of processing time received by each job j up to time t is uniquely determined. In particular, this holds for jobs in S, for which we have $x_j^{\mathcal{A}}(t) = \pi_j$, for all $j \in S$.
$\Rightarrow$: Let j be in S. Then, by Corollary 3, the time spent by $\mathcal{A}$ on j up to time t is π_j. Since j is active at time t, $p_j > x_j^{\mathcal{A}}(t) = \pi_j$.
$\Leftarrow$: Let I' denote the instance such that $\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}}$ and $p_{jI'} > \pi_j$ for all $j \in S$. Let I denote the instance such that $\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}}$ and $p_{jI} = \pi_j$ for all $j \in S$. Consider the two deterministic schedules corresponding to I and I'. By construction and since MLF is oblivious to the processing times of the jobs, we know that in the schedule corresponding to instance I (i) no job in S is completed before t and (ii) jobs that are not in S have either been completed by time t, or they are yet to be released. Then, by Fact 6, $x_{jI}^{\mathcal{A}}(t) = x_{jI'}^{\mathcal{A}}(t)$ for all j. This implies that $S_{I'}(t) = S$, since $p_{jI'} > \pi_j = x_{jI}^{\mathcal{A}}(t)$, for all $j \in S$, while jobs not in S are either yet to be released, or they have been completed by time t, since by Fact 6 they have received the same amount of processing time in the two schedules. □

Lemma 12. *The variables $Y_j = (X_j^l \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}}))$, $j \in S$, are mutually independent.*

Proof. Let $R \subseteq S$ and let $a_j \in \{0, 1\}$ for each $j \in R$.

$$\begin{aligned}
\mathbf{P}\left[\bigcap_{j\in R} X_j^l = a_j \,\middle|\, (S(t) = S) \cap (\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}})\right] &= \mathbf{P}\left[\bigcap_{j\in R} X_j^l = a_j \,\middle|\, \bigcap_{j\in S}(p_j > \pi_j) \cap (\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}})\right] \\
&= \mathbf{P}\left[\bigcap_{j\in R} p_j \in I_j \,\middle|\, \bigcap_{j\in S}(p_j > \pi_j) \cap (\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}})\right],
\end{aligned}$$

where the first equality follows from Lemma 11, and I_j denotes the union of intervals such that $(X_j^l = a_j)$ holds.

$$\mathbf{P}\left[\bigcap_{j\in R} X_j^l = a_j \,\middle|\, (S(t) = S) \cap (\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}})\right] = \frac{\mathbf{P}\left[\bigcap_{j\in R}(p_j \in I_j) \cap \bigcap_{j\in S}(p_j > \pi_j) \cap (\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}})\right]}{\mathbf{P}\left[\bigcap_{j\in S}(p_j > \pi_j) \cap (\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}})\right]}$$
$$= \frac{\mathbf{P}\left[\bigcap_{j\in R}(p_j \in I'_j) \cap \bigcap_{j\in S\setminus R}(p_j > \pi_j) \cap (\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}})\right]}{\mathbf{P}\left[\bigcap_{j\in S}(p_j > \pi_j) \cap (\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}})\right]},$$

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

$$\mathbf{P}\left[\bigcap_{j\in R} X_j^l = a_j \,\middle|\, (S(t) = S) \cap (\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}})\right] = \frac{\prod_{j\in R}\mathbf{P}[p_j \in I'_j]\,\mathbf{P}\left[\bigcap_{j\in S\setminus R}(p_j > \pi_j) \cap (\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}})\right]}{\prod_{j\in R}\mathbf{P}[p_j > \pi_j]\,\mathbf{P}\left[\bigcap_{j\in S\setminus R}(p_j > \pi_j) \cap (\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}})\right]}$$
$$= \prod_{j\in R}\frac{\mathbf{P}[p_j \in I'_j]}{\mathbf{P}[p_j > \pi_j]} = \prod_{j\in R}\mathbf{P}[X_j^l = a_j \,|\, p_j > \pi_j]$$

The above equality holds for any subset $R \subseteq S$. In particular, for a single job $j \in R$,

$$\mathbf{P}[X_j^l = a_j \,|\, (S(t) = S) \cap (\mathbf{p}_{\bar{S}} = \mathbf{x}_{\bar{S}})] = \mathbf{P}[X_j^l = a_j \,|\, p_j > \pi_j]\,.$$

□

Lemma 8. *For any $S \subseteq J$, at any time t, $\mathbf{P}[\delta^l(t) < \frac{1}{2}\alpha\delta(t) \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}})] \leq e^{-\frac{\alpha|S|}{8}}$.*

Proof. By Lemma 12, the random variables $Y_j = (X_j^l \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}}))$, $j \in S$, are independent. Moreover, by Lemma 7 we have $\mathbf{E}[\delta^l(t) \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}})] \geq \alpha|S|$. Applying the standard Chernoff bound to $(\delta^l(t) \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}})) = (\sum_{j\in J} Y_j)$, we obtain

$$\begin{aligned}\mathbf{P}[\delta^l(t) < \tfrac{1}{2}\alpha\delta(t) \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}})] &= \mathbf{P}[\delta^l(t) < \tfrac{1}{2}\alpha|S| \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}})] \\ &\leq \mathbf{P}[\delta^l(t) < \tfrac{1}{2}\mathbf{E}[\delta^l(t) \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}})] \,|\, \mathcal{F}(t, S, \mathbf{x}_{\bar{S}})] \leq e^{-\frac{\alpha|S|}{8}}.\end{aligned}$$

□

I Proofs of Lower Bounds

In this part of the appendix, we prove the lower bounds on the smoothed competitive ratio. We advise the reader to first read the proof for the adaptive adversary since this bound is more intuitive. We present the lower bounds in the order in which they appeared in the paper.

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

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 $\mathcal{OPT}$ 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 $\min(N - 2^{K/2}, \lfloor 2^K/(3 \cdot 2^k) \rfloor) \geq N - 2^{K/2}$ jobs, i.e., all these jobs. In case (ii), by not processing job j^*, $\mathcal{S}$ completes at least $\min(N-1, \lfloor (2^K - 2^{k+1})/(3 \cdot 2^k) \rfloor) \geq N - 2^{K/2}$ 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 $\bar{p}_j = 1$ 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 $\mathcal{A}$ 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_j's are independently distributed random variables with zero mean. Define $S_0 = 0$ and $S_i = \sum_{1 \leq j \leq i} X_j$, for $i = 1, \ldots, L$. Applying Kolmogorov's inequality, 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 $I_i = \max(I_{i-1}, i\mu - \sum_{1 \leq j \leq i} p_{N+j})$, where $I_0 = 0$. 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 $\mathcal{A}$ during the second phase.

Lemma 13. *With probability at least $\frac{2}{3}$, at any time $t \in [\hat{t}, \hat{t} + \tilde{t}]$: $\delta^{\mathcal{A}}(t) \geq N - \frac{1}{2}\sqrt{L} - 1$.*

Proof. $\mathcal{A}$ 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, $\mathcal{A}$ 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 $\mathcal{A}$ 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 14. *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

$$\sum_{1 \leq j \leq i} p_j - (i\mu - I_i)$$

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

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

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

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

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, $\mathcal{S}$ 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

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

□

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

$$\begin{aligned} F^{\mathcal{A}} &\geq (N - \sqrt{L}/2 - 1)\tilde{t}, \\ F^{\mathcal{S}} &\leq N\hat{t} + (2^{K/2} + 2\sqrt{L} + 2)\tilde{t} + (2^{K/2} + 2\sqrt{L} + 2)(3N2^k + 2\mu\sqrt{L}), \end{aligned}$$

where the contribution of the period after time $\hat{t}+\tilde{t}$, for $\mathcal{S}$ 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 $F^{\mathcal{A}}$ and $F^{\mathcal{S}}$, we note that from the upper bounds on N and $\hat{t}$ it follows that $N\hat{t} \leq 2(2^{K/2}+2\sqrt{L}+2)\mu L$. Moreover, we know from the definition of N and μ that $3N2^k + 2\mu\sqrt{L} \leq 8\mu L$. Hence, by restricting $k \leq K/3$, we have that

$$\begin{aligned}\mathbf{E}\left[\frac{F^{\mathcal{A}}}{F^{\mathcal{OPT}}}\right] &= \Omega\left(\frac{N-\sqrt{L}/2-1}{2^{K/2}+2\sqrt{L}+2}\right)\\ &= \Omega\left(\frac{2^{K-k}+2^{K/2}-2^{5K/6-k/2}}{2^{5K/6-k/2}}\right)\\ &= \Omega\left(2^{K/6-k/2}\right).\end{aligned}$$

□

Theorem 3. *Any deterministic algorithm $\mathcal{A}$ has smoothed competitive ratio $\Omega(2^{K-k})$ against an adaptive adversary in the partial bit randomization smoothening 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 $\mathcal{A}$ 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 $x_j^{\mathcal{A}}(\hat{t})$ denote the amount of time spent by algorithm $\mathcal{A}$ on job j up to time $\hat{t}$. We fix the initial processing time of each job j to $\bar{p}_j = x_j^{\mathcal{A}}(\hat{t}) + 2^{k+1}$. Note that after smoothening the $\bar{p}_j$'s we have $x_j^{\mathcal{A}}(\hat{t}) + 2^k < p_j < x_j^{\mathcal{A}}(\hat{t}) + 3\cdot 2^k$ for each j. That is, each job has a remaining processing time between 2^k and $3\cdot 2^k$. Therefore, $\mathcal{A}$ will not complete any job at time $\hat{t}$, i.e., $\delta^{\mathcal{A}}(\hat{t}) = N$.

Consider the optimal algorithm $\mathcal{OPT}$. If $\mathcal{OPT}$ 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 $\mathcal{OPT}$ until time $\hat{t}$, i.e., $\delta^{\mathcal{OPT}}(\hat{t}) = 1$.

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 $\bar{p}_j = 1$ and therefore its smoothed processing time satisfies $p_j \leq 2^k$.

$\mathcal{OPT}$ will then complete every job released in the second phase before the next one is released. The optimal strategy for $\mathcal{A}$ 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 $\mathcal{A}$ and $\mathcal{OPT}$. Since in the second phase $\mathcal{A}$ has $\Omega(N)$ jobs and $\mathcal{OPT}$ has $O(1)$ jobs in the system, we obtain a competitive ratio of $\Omega(N) = \Omega(2^{K-k})$. □

Consider a function $\varphi : \mathbb{R}^+ \to \mathbb{R}^+$, which is continuous and non-decreasing. The symmetric smoothening model according to φ smoothens the initial processing times as follows.

$$p_j = \max(1, \bar{p}_j + \epsilon_j),$$

where ϵ_j is chosen randomly from $[-\varphi(\bar{p}_j)/2, \varphi(\bar{p}_j)/2]$ according to some probability distribution f.

Theorem 4. *Let $\varphi : \mathbb{R}^+ \to \mathbb{R}^+$ be function such that $\varphi(y) < 2^{K-2}$ for all y, and let $a \geq 1$ such that there exist $x \in \mathbb{R}^+$ satisfying $x + \varphi(x)/2 = 2^{K-1} + a$. Then, there exists an $\Omega(2^K/a)$ lower bound on the smoothed competitive ratio of MLF against an oblivious adversary in the symmetric smoothening model according to φ.*

Proof. 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 completing all phase-2 jobs, $\mathcal{S}$ finishes the remaining phase-1 jobs. We upper bound $\mathcal{OPT}$ by $\mathcal{S}$. To prove the theorem, we show that with constant probability $F^{\mathcal{A}}/F^{\mathcal{S}} = \Omega(2^K/a)$. Then $\mathbf{E}[F^{\mathcal{A}}/F^{\mathcal{OPT}}] = \Omega(2^K/a)$. Without loss of generality, we assume that $K \geq 3$, and we define $L = \varphi(x)$.

Phase 1: At time $t = 0$, $M = 8\max(L^3/2^K, 1)$ jobs are released with initial processing time $\bar{p}_1 = x$ and then every $\bar{p}_1$ time units one job with 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 $i\bar{p}_1$, for $i = 0, 1, \dots, N - M$. Define $S_0 = 0$ and $S_i = S_{i-1} + \epsilon_i = \sum_{j=1}^{i} \epsilon_j$, for $i = 1, \dots, N$. As $\mathbf{E}[\epsilon_j] = 0$ and all ϵ_j are drawn independently, we have that $\mathbf{E}[S_i] = 0$ and $\mathbf{E}[S_i^2] = iL^2/12$, for all $i = 0, 1, \dots, N$. Applying Kolmogorov's inequality, we obtain

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

Hence, we have with probability at least 11/12, that for all $i = 0, 1, \dots, N - M$,

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

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

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

$$\begin{aligned}
T_1(\lfloor t/\bar{p}_1 \rfloor) - t &\geq (\lfloor t/\bar{p}_1 \rfloor + M)\bar{p}_1 - L\sqrt{N} - t \\
&\geq t - 1 + M\bar{p}_1 - L\sqrt{N} - t \\
&\geq M2^{K-2} - L\sqrt{N} - 1 \\
&\geq 2\max(L^3, 2^K) - \max(L^3, 2^K) - 1 > 0.
\end{aligned}$$

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

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

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

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 a Chernoff bound, we know that with probability at least $1 - e^{aN/8L} \geq (e-1)/e$, 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 $(e-1)/e - 1/12$.

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

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

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 that the total remaining processing time is at least $L\sqrt{N} - 1$. Hence, at this time the number of alive jobs in the schedule of MLF is at least $(L\sqrt{N} - 1)/a$ and also

$$\delta^{\mathcal{A}}(\hat{t}) \geq (L\sqrt{N} - 1)/a.$$

Phase 2: At time $\hat{t}$, M jobs with $\bar{p}_2 = 2^{K-2}$ are released and then every $\bar{p}_2$ time units one job with the same $\bar{p}_2$ is released. The total number of jobs released in this phase is $2N$. Note that 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 release at or before time $\hat{t} + i\bar{p}_2$. Applying Kolmogorov's inequality yields that with probability at least $11/12$, we have that

$$(i + M)\bar{p}_2 - L\sqrt{2N} \leq T_2(i) \leq (i + M)\bar{p}_2 + L\sqrt{2N}. \tag{6}$$

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

Using the same arguments as before, we now show that MLF continuously processes phase-2 jobs until time $\bar{t} = \hat{t} + (2N - M + 1)\bar{p}_2$. 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

$$\begin{aligned}
T_2(\lfloor (t - \hat{t})/\bar{p}_2 \rfloor) - (t - \hat{t}) &\geq (\lfloor (t - \hat{t})/\bar{p}_2 \rfloor + M)\bar{p}_2 - L\sqrt{2N} - (t - \hat{t}) \\
&\geq M\bar{p}_2 - L\sqrt{2N} - 1 \\
&\geq M2^{K-2} - L\sqrt{2N} - 1 \\
&\geq 2\max(L^3, 2^K) - \sqrt{2}\max(L^3, 2^K) - 1 > 0.
\end{aligned}$$

Thus,

$$\delta^{\mathcal{A}}(t) \geq \frac{aN}{2L} \quad t \in [\hat{t}, \bar{t}),$$

and

$$F^{\mathcal{A}} = \Omega\left(\frac{aN}{2L}(2N - M + 1)\bar{p}_2\right),$$

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

$$\delta^{\mathcal{A}}(t) \geq (L\sqrt{N} - 1)/a, \quad t \in [\hat{t}, \bar{t}),$$

and

$$F^{\mathcal{A}} = \Omega(L\sqrt{N}(2N - M + 1)\bar{p}_2).$$

Moreover, using the same argumentation as for phase 1, we know that during $[\hat{t}, \bar{t})$, $\mathcal{S}$ has at most $(2+\sqrt{2})M+1$ second phase jobs in its system. Hence,

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

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

$$\begin{aligned} L\sqrt{N} + L\sqrt{2N} &= \frac{1+\sqrt{2}}{8} M 2^K \\ &\leq \frac{1}{2}(1+\sqrt{2}) M \bar{p}_2 \\ &\leq \frac{1}{2}(1+\sqrt{2})(2N - M + 1)\bar{p}_2. \end{aligned}$$

Hence,

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

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

$$F^{\mathcal{A}}/F^{\mathcal{S}} = \Omega(L\sqrt{N}/M) = \Omega(2^K/a),$$

or

$$F^{\mathcal{A}}/F^{\mathcal{S}} = \Omega((aN/2L)/M) = \Omega(a2^K).$$

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

$$F^{\mathcal{A}}/F^{\mathcal{S}} = \Omega(L\sqrt{N}/M) = \Omega(2^K/a),$$

or

$$F^{\mathcal{A}}/F^{\mathcal{S}} = \Omega((aN/2L)/M) = \Omega(a2^K).$$

Since the probability that (5), (6), and the bound on the number of class K jobs hold is constant and $a \geq 1$, we have

$$\mathbf{E}\left[\frac{F^{\mathcal{A}}}{F^{\mathcal{OPT}}}\right] = \Omega(2^K/a).$$

□

I N F O R M A T I K

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-2003-NWG2-002	F. Eisenbrand	Fast integer programming in fixed dimension
MPI-I-2003-NWG2-001	L.S. Chandran, C.R. Subramanian	Girth and Treewidth
MPI-I-2003-4-009	N. Zakaria	FaceSketch: An Interface for Sketching and Coloring Cartoon Faces
MPI-I-2003-4-008	C. Roessl, I. Ivrissimtzis, H. Seidel	Tree-based triangle mesh connectivity encoding
MPI-I-2003-4-007	I. Ivrissimtzis, W. Jeong, H. Seidel	Neural Meshes: Statistical Learning Methods in Surface Reconstruction
MPI-I-2003-4-006	C. Roessl, F. Zeilfelder, G. Nürnberger, H. Seidel	Visualization of Volume Data with Quadratic Super Splines
MPI-I-2003-4-005	T. Hangelbroek, G. Nürnberger, C. Roessl, H.S. Seidel, F. Zeilfelder	The Dimension of C^1 Splines of Arbitrary Degree on a Tetrahedral Partition
MPI-I-2003-4-004	P. Bekaert, P. Slusallek, R. Cools, V. Havran, H. Seidel	A custom designed density estimation method for light transport
MPI-I-2003-4-003	R. Zayer, C. Roessl, H. Seidel	Convex Boundary Angle Based Flattening
MPI-I-2003-4-002	C. Theobalt, M. Li, M. Magnor, H. Seidel	A Flexible and Versatile Studio for Synchronized Multi-view Video Recording
MPI-I-2003-4-001	M. Tarini, H.P.A. Lensch, M. Goesele, H. Seidel	3D Acquisition of Mirroring Objects
MPI-I-2003-2-003	Y. Kazakov, H. Nivelle	Subsumption of concepts in DL $\mathcal{FL}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete
MPI-I-2003-2-002	M. Jaeger	A Representation Theorem and Applications to Measure Selection and Noninformative Priors
MPI-I-2003-2-001	P. Maier	Compositional Circular Assume-Guarantee Rules Cannot Be Sound And Complete
MPI-I-2003-1-015	A. Kovács	Sum-Multicoloring on Paths
MPI-I-2003-1-014	G. Schäfer	Average Case and Smoothed Competitive Analysis of the Multi-Level Feedback Algorithm
MPI-I-2003-1-013	I. Katriel, S. Thiel	Fast Bound Consistency for the Global Cardinality Constraint
MPI-I-2003-1-012	D. Fotakis, R. Pagh, P. Sanders, P. Spirakis	Space Efficient Hash Tables with Worst Case Constant Access Time
MPI-I-2003-1-011	P. Krysta, A. Czumaj, B. Voecking	Selfish Traffic Allocation for Server Farms
MPI-I-2003-1-010	H. Tamaki	A linear time heuristic for the branch-decomposition of planar graphs

MPI-I-2003-1-009	B. Csaba	On the Bollobás – Eldridge conjecture for bipartite graphs
MPI-I-2003-1-008	P. Sanders	Soon to be published
MPI-I-2003-1-007	H. Tamaki	Alternating cycles contribution: a strategy of tour-merging for the traveling salesman problem
MPI-I-2003-1-006	M. Dietzfelbinger, H. Tamaki	On the probability of rendezvous in graphs
MPI-I-2003-1-005	M. Dietzfelbinger, P. Woelfel	Almost Random Graphs with Simple Hash Functions
MPI-I-2003-1-004	E. Althaus, T. Polzin, S.V. Daneshmand	Improving Linear Programming Approaches for the Steiner Tree Problem
MPI-I-2003-1-003	R. Beier, B. Vöcking	Random Knapsack in Expected Polynomial Time
MPI-I-2003-1-002	P. Krysta, P. Sanders, B. Vöcking	Scheduling and Traffic Allocation for Tasks with Bounded Splittability
MPI-I-2003-1-001	P. Sanders, R. Dementiev	Asynchronous Parallel Disk Sorting
MPI-I-2002-4-002	F. Drago, W. Martens, K. Myszkowski, H. Seidel	Perceptual Evaluation of Tone Mapping Operators with Regard to Similarity and Preference
MPI-I-2002-4-001	M. Goesele, J. Kautz, J. Lang, H.P.A. Lensch, H. Seidel	Tutorial Notes ACM SM 02 A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2002-2-008	W. Charatonik, J. Talbot	Atomic Set Constraints with Projection
MPI-I-2002-2-007	W. Charatonik, H. Ganzinger	Symposium on the Effectiveness of Logic in Computer Science in Honour of Moshe Vardi
MPI-I-2002-1-008	P. Sanders, J.L. Träff	The Factor Algorithm for All-to-all Communication on Clusters of SMP Nodes
MPI-I-2002-1-005	M. Hoefer	Performance of heuristic and approximation algorithms for the uncapacitated facility location problem
MPI-I-2002-1-004	S. Hert, T. Polzin, L. Kettner, G. Schäfer	Exp Lab A Tool Set for Computational Experiments
MPI-I-2002-1-003	I. Katriel, P. Sanders, J.L. Träff	A Practical Minimum Scanning Tree Algorithm Using the Cycle Property
MPI-I-2002-1-002	F. Grandoni	Incrementally maintaining the number of l-cliques
MPI-I-2002-1-001	T. Polzin, S. Vahdati	Using (sub)graphs of small width for solving the Steiner problem
MPI-I-2001-4-005	H.P.A. Lensch, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2001-4-004	S.W. Choi, H. Seidel	Linear One-sided Stability of MAT for Weakly Injective Domain
MPI-I-2001-4-003	K. Daubert, W. Heidrich, J. Kautz, J. Dischler, H. Seidel	Efficient Light Transport Using Precomputed Visibility
MPI-I-2001-4-002	H.P.A. Lensch, J. Kautz, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing, Transmission, and Interactive Display of High Quality 3D Models on the Web
MPI-I-2001-4-001	H.P.A. Lensch, J. Kautz, M. Goesele, W. Heidrich, H. Seidel	Image-Based Reconstruction of Spatially Varying Materials
MPI-I-2001-2-006	H. Nivelle, S. Schulz	Proceeding of the Second International Workshop of the Implementation of Logics
MPI-I-2001-2-005	V. Sofronie-Stokkermans	Resolution-based decision procedures for the universal theory of some classes of distributive lattices with operators
MPI-I-2001-2-004	H. de Nivelle	Translation of Resolution Proofs into Higher Order Natural Deduction using Type Theory
MPI-I-2001-2-003	S. Vorobyov	Experiments with Iterative Improvement Algorithms on Completely Unimodel Hypercubes
MPI-I-2001-2-002	P. Maier	A Set-Theoretic Framework for Assume-Guarantee Reasoning
MPI-I-2001-2-001	U. Waldmann	Superposition and Chaining for Totally Ordered Divisible Abelian Groups
MPI-I-2001-1-007	T. Polzin, S. Vahdati	Extending Reduction Techniques for the Steiner Tree Problem: A Combination of Alternative-and Bound-Based Approaches

MPI-I-2001-1-006	T. Polzin, S. Vahdati	Partitioning Techniques for the Steiner Problem
MPI-I-2001-1-005	T. Polzin, S. Vahdati	On Steiner Trees and Minimum Spanning Trees in Hypergraphs
MPI-I-2001-1-004	S. Hert, M. Hoffmann, L. Kettner, S. Pion, M. Seel	An Adaptable and Extensible Geometry Kernel
MPI-I-2001-1-003	M. Seel	Implementation of Planar Nef Polyhedra