Modeling Locality: A Probabilistic Analysis of LRU and FWF *

Luca Becchetti[1]

University of Rome "La Sapienza",
becchett@dis.uniroma1.it

Abstract. In this paper we explore the effects of locality on the performance of paging algorithms. Traditional competitive analysis fails to explain important properties of paging assessed by practical experience. In particular, the competitive ratios of paging algorithms that are known to be efficient in practice (e.g. LRU) are as poor as those of naive heuristics (e.g. FWF). It has been recognized that the main reason for these discrepancies lies in an unsatisfactory modelling of locality of reference exhibited by real request sequences.
Following [14], we explicitely address this issue, proposing an adversarial model in which the probability of requesting a page is also a function of the page's age. In this way, our approach allows to capture the effect of locality of reference. We consider several families of distributions and we prove that the competitive ratio of LRU becomes constant as locality increases, as expected. This result is strengthened when the distribution satisfies a further concavity/convexity property: in this case, the competitive ratio of LRU is always constant.
We also propose a family of distributions parametrized by locality of reference and we prove that the performance of FWF rapidly degrades as locality increases, while the converse happens for LRU.
We think, our results provide one contribution to explaining the behaviours of these algorithms in practice.

1 Introduction

Paging is the problem of managing a two-level memory consisting of a first level fast memory or *cache* and a second level slower memory, both divided into pages of equal, fixed size. We assume that the slow memory contains a universe $\mathcal{P} = \{p_1, \ldots, p_M\}$ of pages that can be requested, while the cache can host a fixed number k of pages. Normally, $M \gg k$.

The input to a paging algorithm is a sequence of on-line requests $\sigma = \{\sigma(1), \ldots, \sigma(n)\}$, each specifying a page to access. The i-th request is said to have *index* i. If the page is in main memory the access operation costs nothing. Otherwise, a page fault occurs: the requested page has to be brought from main memory into cache in order to be accessed. If the cache is full, one of the pages

* Partially supported by Ialian MIUR Project Firb Web Minds, the EU project AMORE the EU projects APPOL II and COSIN.

in cache has to be evicted, in order to make room for the new one. When a page fault occurs, the cost of the access operation is 1. Each page has to be served before the next request is presented. The goal is minimizing the total cost, i.e. the number of page faults. We use $|\sigma|$ to denote the length of σ.

Given a sequence σ, the *cache state* $\mathcal{C}(\sigma)$ is the set of pages in cache after σ has been served. The output of a paging algorithm corresponding to request sequence σ is the sequence $\mathcal{C}(\sigma_1), \ldots, \mathcal{C}(\sigma_{|\sigma|})$ of cache states, with $\sigma_1 = \sigma(1)$ and $\sigma_i = \sigma_{i-1}\sigma(i)$ for $i > 1$.

Paging algorithms. The core of a paging algorithm is the strategy according to which pages are evicted in the presence of faults. In this paper we analyze the performance of two well known heuristics for page eviction, namely LRU (Least Recently Used) and FWF (Flush When Full). The former is the best known algorithm for virtual memory management in operating systems [18] and has found application in web caching as well. The latter has mostly theoretical interest. For the sake of the analysis we are also interested in the off-line optimum. A well known optimal algorithm for paging is LFD (Longest Forward Distance). These algorithms are described below.

- **Longest Forward Distance (LFD):** Replace the page whose next request is latest. LFD is an optimal off-line algorithm and was proposed in an early work by Belady [2].
- **Least Recently Used (LRU):** When a page fault occurs, replace the page whose most recent request was earliest.
- **Flush When Full (FWF):** Assume the cache is initially empty. Three cases may occur by a request: i) the page requested is in cache: in this case the request is served at no extra cost; ii) a page fault occurs and the cache is not full: in this case the page is brought into cache, no page is evicted; iii) a page fault occurs and the cache is full: in this case, *all* pages in the cache are evicted, i.e. the cache is flushed. Note that both in case ii) and iii) the cost incurred by the algorithm is 1.

Both LRU and FWF belong to the class of *marking algorithms*. Roughly speaking, marking algorithms try to keep recently requested pages in cache, under the assumption that pages that have been requested in the recent past are more likely to be requested in the near future. Many marking algorithms have been proposed in practice, the most prominent ones are described in [3, 18].

Contribution of this paper. Theory shows that LRU and the naive FWF heuristic have the same competitive ratio, practice provides pretty different results [20]: LRU's performance is almost optimal on most instances of practical interest, whereas FWF often behaves poorly, as intuition suggests. This discrepancy is a consequence of the worst case approach followed by competitive analysis, which fails to capture the phenomenon of locality [18].

In this paper, we extend the *diffuse adversary* introduced by Koutsoupias and Papadimitriou [14]. In their model, the adversary is constrained to generate the input sequence according to a given family of probability distributions. We extend the work in [14] and [24, 22] in the following way: i) We propose a new framework in which the probability that some page p is requested, conditioned to any request sequence prior to p, also depends on how recently p was last requested; ii) We prove that, under a very large class of distributions, LRU's competitive ratio rapidly tends to become constant as locality increases, as observed in practice; iii) We assess the poor behaviour of FWF, by proposing a family of distributions parametrized by locality of reference and proving that FWF's performance degrades as locality increases. Together with the former ones, this result sharply discriminates bertween the behaviours of LRU and FWF, as observed in practice.

Related results. Sleator and Tarjan [17] introduced competitive analysis and were the first to show that FIFO and LRU are k-competitive. They also proved that this is the lower bound for any paging algorithm. Successively, Young and Torng [23, 19] proved that all paging algorithms falling in two broad classes, including FIFO, LRU and FWF, are k-competitive. Torng [19] also proves that all marking algorithms are $\Theta(k)$-competitive[1].

A considerable amount of research was devoted to overcoming the apparent discrepancies between theory and practice. Due to lack of space, we give a non-chronological overview of research in the field, giving emphasis to contributions that are closer to our approach in spirit.

The results of [14] where extended by Young [24, 22], who proved that the competitive ratio of a subclass of marking algorithms (not including FWF) rapidly becomes constant as $\epsilon = O(1/k)$, is $\Theta(\log k)$ when $\epsilon = 1/k$ and rapidly becomes $\Omega(k)$ as $\epsilon = \Omega(1/k)$. He also proved that, under this adversary, this is asymptotically the lower bound for any deterministic on-line algorithm. The role of randomization was explored by Fiat et al. [9], who proposed a randomized paging algorithm MARK that is H_k-competitive, a result which is tight.

Borodin et al. [4] assume that pages are vertices of an underlying graph, called *access graph*. Locality is modelled by constraining request sequences to correspond to simple walks in the access graph. The same model was used and extended in subsequent works [12, 10, 15, 4, 15] and it was extended in [8] to multi-pointer walks in the access graph.

The work of [13] is similar in spirit, although the approach is probabilistic: the authors assume that the request sequence is generated according to a Markov chain M. They consider fault rate, i.e. the long term frequency of page faults with respect to M.

A deterministic approach based on the working set of Denning [7] is proposed by Albers et al. [1].

Young [20, 21] introduces the notion of loose competitiveness. A paging algorithm A is loosely $c(k)$-competitive if, for any request sequence, for all but a small fraction of the possible values for k, the cost of the algorithm is within $c(k)$ times the optimum for every request sequence. The author proves [20] that a broad class of algorithms, including marking algorithms, is loosely $\Theta(\log k)$-competitive.

[1] Torng's result is actually more general, but this is the aspect that is relevant to the topics discussed in this paper.

Finally, Boyar et al. [6] analyze the relative worst order ratios of some prominent paging algorithms, including LRU, FWF and MARK. The worst order ratio of two algorithms A and B, initially proposed for bin packing [5] is defined as the ratio of their worst case costs, i.e. their worst case number of faults in the case of paging, over request sequences of the same length.

Roadmap. This paper is organized as follows: In Section 2 we propose our model and discuss some preliminaries. In Section 3 we provide a diffuse adversary analysis of LRU, where the distributions allowed to the adversary are used to model temporal locality. We prove that LRU is constant competitive with respect to a wide class of distributions. In Section 4 we prove that, under the same diffuse adversary, the performance of FWF is extremely poor. For the sake of brevity and readibility, many proofs are omitted and will be presented in the full version of the paper.

2 Model and preliminaries

The *diffuse adversary* model was proposed by Koutsoupias and Papadimitriou [14]. It removes the overly pessimistic assumption that we know nothing about the distribution according to which the input is generated, instead assuming that it is member of some known family Δ. We say that algorithm A is c-competitive against the Δ-diffuse adversary if, for every distribution $D \in \Delta$ over the set of possible input sequences,

$$\mathbf{E}_D[A(\sigma)] \leq c\mathbf{E}_D[\mathrm{OPT}(\sigma)] + b,$$

where σ is generated according to D and b is a constant. The *competitive ratio* of A against the Δ-diffuse adversary is the minimum c such that the condition above holds for every $D \in \Delta$. In the sequel, we drop D from the expression of the expectation when clear from context.

The request sequence is a stochastic process. Without loss of generality, in the following we consider request sequences of length n. Considered any request sequence σ and its i-th request $\sigma(i)$, we call $\sigma(1)\cdots\sigma(i-1)$ and $\sigma(i+1)\cdots\sigma(n)$ respectively the *prefix* and the *postfix* of $\sigma(i)$. We define the following random variables: If $i \leq n$, $\Gamma(i)$ denotes the i-th page requested. If $i \leq n$, $\overline{\Gamma}(i)$ denotes the prefix of the i-th request.

2.1 Distribution of page references

Similarly to [14] and [24], the family of distributions we consider is completely defined by the probability $\mathbf{P}[p \,|\, \sigma]$ that p is requested provided σ is the prefix, for every $p \in \mathcal{P}$ and for every possible prefix σ that satisfies $|\sigma| \leq n-1$. In particular, we assume that $\mathbf{P}[p \,|\, \sigma]$ belongs to some family Δ. The distributions we consider are intended to model locality of page references. The following definition will be useful to define some of them.

Definition 1. *Consider a monotone, non increasing function $f : I \rightarrow [0,1]$, where $I = \{1,\ldots,N\}$ for some positive integer N. We say that f is* **concave** *in I if, for all $i = 2,\ldots,N-1 : f(i-1) - f(i) \leq f(i) - f(i+1)$. f is* **convex** *in I if, for all $i = 2,\ldots,N-1 : f(i-1) - f(i) \geq f(i) - f(i+1)$.*

Assume σ is the request sequence observed so far. Provided both pages p_{j_1} and p_{j_2} appear in request sequence σ, we say that p_{j_1} is *younger* than p_{j_2} if the last request for p_{j_1} in σ occurs after the last request for p_{j_2}, while we say that p_{j_1} is *older* than p_{j_2} otherwise. The age of any page p_j with respect to σ is then defined as follows: $\mathbf{age}(p_j, \sigma) = l$, if p_j is the l-th most recently accessed page, while $\mathbf{age}(p_j, \sigma) = \infty$ if p_j does not appear in σ. For the sake of simplicity, we assume in the sequel that $\mathbf{age}(p_j, \sigma) < \infty$ for every p_j. This assumption can be easily removed by slightly modifying the definitions that follow. This is shown in the full version of the paper, for brevity. We now define, for every prefix sequence σ, a total order $\mathcal{F}(\sigma)$ on $\mathcal{P}$ by ordering pages according to their relative ages. More precisely, for any pages p_{j_1}, p_{j_2}, $p_{j_1} \prec_{\mathcal{F}(\sigma)} p_{j_2}$ if and only if $\mathbf{age}(p_{j_1}, \sigma) < \mathbf{age}(p_{j_2}, \sigma)$. $\mathcal{F}(\sigma)$ obviously implies a one-to-one correspondence $f_\sigma : \mathcal{P} \rightarrow \{1,\ldots,M\}$, with $f_\sigma(p_j) = l$ if, given σ, p_j is the l-th most recently accessed page. Conversely, $\mathbf{P}[p\,|\,\sigma]$ may be regarded as a distribution over $\{1,\ldots,M\}$ by defining $\overline{\mathbf{P}}[x\,|\,\sigma] = \mathbf{P}[f_\sigma^{-1}(x)\,|\,\sigma]$, for every $x = 1,\ldots,M$. So, for instance, by saying that $\mathbf{P}[p\,|\,\sigma]$ has average value μ, we mean that $\overline{\mathbf{E}}[x\,|\,\sigma] = \sum_{x=1}^{M} x\overline{\mathbf{P}}[x\,|\,\sigma] = \mu$.

2.2 Diffuse adversaries

We next define three families of probability distributions according to which, given any prefix σ, the next page request can be generated.

Definition 2. *$\mathcal{G}(\tau)$ is the family of probability distributions over $\{1,\ldots,M\}$ that have expected value at most $k/2$ and standard deviation τ. $\mathcal{D}_1$ (resp. $\mathcal{D}_2$) are the families of distributions that are monotone, non-increasing and concave (resp. convex) in $\{1,\ldots,M\}$.*

The first property of Definition 2 models the well known folklore that "90 percent of requests is for about 10 percent of the pages".

Figure 2.2 shows an example in which $\mathbf{P}[p\,|\,\sigma] \in \mathcal{D}_2$. The y coordinate is $\overline{\mathbf{P}}[x\,|\,]\sigma]$. Observe that the l-th element along the x-axis represents the l-th most recently requested page and thus the first k elements correpond to the set of recent pages. In the case of LRU these are also the pages in the cache of the algorithm.

We next describe how the adversary generates request sequences. Assumed a prefix request sequence σ of length $i < n$ has been generated, the adversary picks a distribution $D(i)$ out of some family Δ and it chooses the next page to request ($\sigma(i+1)$) according to $D(i)$. Observe that, in general, $D(i_1) \neq D(i_2)$ for $i_1 \neq i_2$. We define different adversaries, according to how $D(i)$ is chosen:

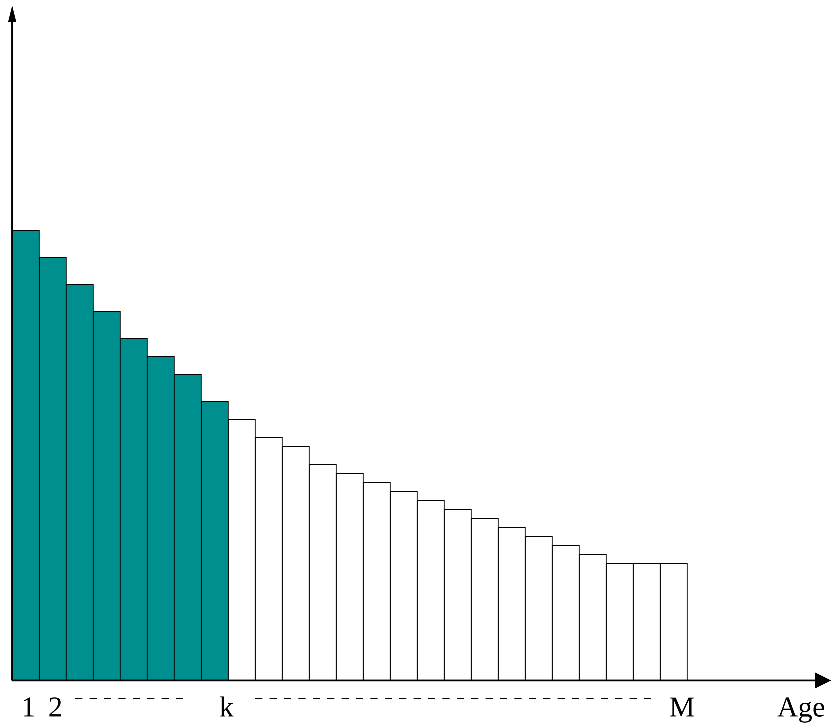

Fig. 1. This picture represents the case in which $\mathbf{P}[p \mid \sigma] \in \mathcal{D}_2$.

Definition 3. *An adversary is concentrated (respectively concave and convex) if $D(i) \in \mathcal{G}(\tau)$ (respectively $\mathcal{D}_1$ and $\mathcal{D}_2$) for all i. If $D(i) = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{G}(\tau)$ for some τ and for all i we speak of a general adversary.*

Observe that the general adversary resumes all possible subcases.

2.3 Preliminary results

We partition each request sequence into *phases* in the usual way [3]. In particular, phases are defined inductively as follows: i) phase 1 starts with the first request and completes right before the $(k+1)$-th *distinct* request has been presented; ii) assuming phase $i-1$, phase i starts with the $(k+1)$-th distinct request after the start of phase $i-1$ and it completes right before the $(2k+1)$-th distinct request after the start of phase $i-1$. We point out that the last phase of a request sequence might contain less than k distinct requests. However, it has to contain at least one. In the sequel, F is the random variable that denotes the overall number of phases.

We say that a page is *marked* in a phase, if it was already requested during the phase. Otherwise the page is said *unmarked.* A *marking* algorithm never evicts a marked page when a page fault occurs. Observe that both LRU and FWF are marking algorithms. As such, they both achieve the best possible competitve ratio k for on-line paging algorithms [19].

Considered any phase $\ell \geq 1$, we say that a page p_j is *new* in ℓ if it was not requested during phase $\ell-1$. Otherwise we say that p_j is *old*. Also, considered any point of ℓ, any of the k most recently requested pages is said *recent*. Considered any prefix σ, we denote by $\mathcal{N}(\sigma)$ and $\mathcal{R}(\sigma)$ respectively the set of new and recent pages after request subsequence σ has been presented. Observe that both $\mathcal{N}(\sigma)$ and $\mathcal{R}(\sigma)$ do not depend on the paging algorithm. We say that page p_j is *critical*, if p_j is old but it is not currently in cache. Observe that, considered any prefix σ, the sets of in cache and critical pages depend on the particular paging algorithm under consideration. In the sequel, for any paging algorithm A, we denote by $\mathcal{C}_A(\sigma)$ and $\mathcal{O}_A(\sigma)$ respectively the sets of in cache and critical pages. Observe that $\mathcal{C}_A(\sigma) \neq \mathcal{R}(\sigma)$ in general. Also, we write $\mathcal{C}(\sigma)$ for $\mathcal{C}_A(\sigma)$ and $\mathcal{O}(\sigma)$ for $\mathcal{O}_A(\sigma)$ whenever A is clear from context.

W^ℓ and N^ℓ are random variables whose values are respectively the number of requests to critical and new pages during phase ℓ. $W^\ell = N^\ell = 0$ if no ℓ-th phase exists. A request in phase ℓ is said *distinct* if it specifies a page that was not previously requested during the phase. Note that requests for new pages are distinct by definition. We denote by D^ℓ the number of distinct requests during phase ℓ. Observe that, differently from N^ℓ and D^ℓ, W^ℓ depends on the paging algorithm. However, in the rest of this paper the algorithm is always understood from context, hence we do not specify it for these variables. Finally, considered any random variable V defined over all possible request sequences of length n, we denote by V_σ its deterministic value for a particular request sequence σ. The following result is due to Young [24]:

Theorem 1 **([24]).** *For every request sequence σ:*

$$\frac{1}{2}\sum_\ell N_\sigma^\ell \leq OPT(\sigma) \leq \sum_\ell N_\sigma^\ell.$$

In the sequel, we assume that the first request of the first phase is for a new page. We can assume this without loss of generality, since a sequence of requests for recent pages does not change the cache state of LRU and has cost 0. This allows us to state the following fact:

Fact 1 *The first request of every phase is for a new page.*

The following fact, whose proof is straightforward, gives a crucial property of LRU:

Fact 2 $\mathcal{C}_{LRU}(\sigma) = \mathcal{R}(\sigma)$, *for every* σ.

Observe that Fact 2 in general does not hold for other (even marking) algorithms. Set $\ell(n) = \lceil n/k \rceil$. The following fact will be useful in the sequel:

Fact 3 $F \leq \ell(n)$ *deterministically for request sequences of length* n.

3 Analysis of LRU

The main goal of this section is to bound

$$\frac{\mathbf{E}[\mathrm{LRU}(\sigma)]}{\mathbf{E}[\mathrm{OPT}(\sigma)]},$$

when σ is generated by any of the adversaries defined above. We can can express the competitive ratio of LRU as stated in the following lemma:

Lemma 1.

$$\frac{\mathbf{E}[LRU(\sigma)]}{\mathbf{E}[OPT(\sigma)]} \leq 2\frac{\sum_{\ell=1}^{\ell(n)} \mathbf{E}\left[W^{\ell} \,|\, F \geq \ell\right] \mathbf{P}[F \geq \ell]}{\sum_{\ell=1}^{\ell(n)} \mathbf{E}[N^{\ell} \,|\, F \geq \ell] \,\mathbf{P}[F \geq \ell]} + 2. \tag{1}$$

We now introduce the following random variables, for all pairs (ℓ, j), where $j = 1, \ldots, k$: i) $X_j^{\ell} = 1$ if there is an ℓ-th phase, there is a distinct j-th request in the ℓ-th phase and this request is for a critical page, $X_j^{\ell} = 0$ otherwise (i.e. no ℓ-th phase exists or the phase is not complete and it does not contain a j-th distinct request or it does, but this request is not for a critical page); ii) Similarly, $Y_j^{\ell} = 1$ if the $j-th$ distinct request in phase ℓ is to a new page, $Y_j^{\ell} = 0$ otherwise; a consequence of the definitions above is that we have $\sum_{j=1}^{k} X_j^{\ell} = W^{\ell}$ and analogously for N^{ℓ}. Observe, again, that Y_j^{ℓ} does not depend on the paging algorithm considered, while X_j^{ℓ} does. Again, the algorithm is always understood from context.

The rest of our proof goes along the following lines: intuitively, in the general case we have that, as $\overline{\mathbf{P}}[x \,|\, \sigma]$ becomes concentrated (i.e. variance decreases), most requests in the phase are for pages in cache, with the exception of a small number that at some point rapidly decreases with variance. In the other cases, we are able to bound the expected number of requests for old pages with the expected number of requests for new pages. We point out that in all cases the proof crucially relies on Fact 2.

3.1 Convex adversaries

We consider the case in which $D(i) \in \mathcal{D}_2$, for every $i = 1, \ldots, n$. Intuitively, we shall prove that if the probability of requesting a critical page is more than twice the probability of requesting a new page, then the former has to be sufficiently small in absolute terms. For the rest of this subsection, N_j^{ℓ} is a random variable that denotes the number of requests for new pages that are presented before the j-th distinct request of phase ℓ. In particular, $N_j^{\ell} = 0$ if no ℓ-th phase exists while $N_j^{\ell} = N^{\ell}$ if the ℓ-th phase is not complete and it does not contain a j-th distinct request. We now state the following lemma:

Lemma 2. *If the request sequence is generated by a convex adversary and $M \geq 4k$,*

$$\mathbf{P}[X_j^{\ell} = 1 \,|\, F \geq \ell] \leq 2\mathbf{P}[Y_j^{\ell} = 1 \,|\, F \geq \ell] + \frac{16(j-1)}{k^2}\mathbf{E}\left[N^{\ell} \,|\, F \geq \ell\right].$$

Sketch of proof. Observe that the claim trivially holds if $j = 1$, since no critical pages exist in a phase before the first new page is requested. For $j \geq 2$ we can write:

$$\mathbf{P}[X_j^\ell = 1 \,|\, F \geq \ell] = \sum_{x=1}^{j-1} \mathbf{P}[X_j^\ell = 1 \,|\, F \geq \ell \cap N_j^\ell = x] \, \mathbf{P}[N_j^\ell = x \,|\, F \geq \ell] \, .$$

The assumption of convexity implies that, if $\mathbf{P}[X_j^\ell = 1 \,|\, F \geq \ell \cap N_j^\ell = x] > 2\mathbf{P}[Y_j^\ell = 1 \,|\, F \geq \ell \cap N_j^\ell = x]$, then the former is bounded by $16x^2/k^2$. As a consequence:

$$\mathbf{P}[X_j^\ell = 1 \,|\, F \geq \ell \cap N_j^\ell = x] \leq 2\mathbf{P}[Y_j^\ell = 1 \,|\, F \geq \ell \cap N_j^\ell = x] + \frac{16x^2}{k^2}.$$

We continue with:

$$\begin{aligned}
&\mathbf{P}[X_j^\ell = 1 \,|\, F \geq \ell] = \sum_{x=1}^{j-1} \mathbf{P}[X_j^\ell = 1 \,|\, F \geq \ell \cap N_j^\ell = x] \, \mathbf{P}[N_j^\ell = x \,|\, F \geq \ell] \\
&\leq 2\mathbf{P}[Y_j^\ell = 1 \,|\, F \geq \ell] + \frac{16(j-1)}{k^2} \sum_{x=1}^{j-1} x \mathbf{P}[N_j^\ell = x \,|\, F \geq \ell] \\
&\leq 2\mathbf{P}[Y_j^\ell = 1 \,|\, F \geq \ell] + \frac{16(j-1)}{k^2} \mathbf{E}\left[N^\ell \,|\, F \geq \ell\right],
\end{aligned}$$

where the second inequality follows since $x \leq j - 1$, while the third inequality follows since $N_j^\ell \leq N^\ell$ by definition.

We can now write:

$$\mathbf{E}\left[W^\ell \,|\, F \geq \ell\right] \leq 2\mathbf{E}\left[N^\ell \,|\, F \geq \ell\right] + \frac{16}{k^2} \mathbf{E}\left[N^\ell \,|\, F \geq \ell\right] \sum_{j=1}^{k} (j-1) < 10\mathbf{E}\left[N^\ell \,|\, F \geq \ell\right],$$

where the second inequality follows from simple manipulations. Recalling Equation (1) we can finally state the following Theorem:

Theorem 2. *If the request sequence is generated by a convex adversary,*

$$\frac{\mathbf{E}[LRU(\sigma)]}{\mathbf{E}[OPT(\sigma)]} \leq 22.$$

Note that Theorem 2 does not necessarily hold for other marking algorithms, since the proof of Lemma 2 crucially uses Fact 2.

3.2 Concentrated adversary

In this subsection we consider all adversaries such that $D(i) \in \mathcal{G}(\tau)$ for every $i = 1, \ldots, n$. We can now state the following lemma:

Lemma 3. *If the request sequence is generated by a concentrated adversary then, for $j \geq 2$,*

$$\mathbf{P}[X_j^\ell = 1 \cup Y_j^\ell = 1 \,|\, F \geq \ell] \leq \left(\frac{2\tau}{k}\right)^2 .$$

Lemma 3 and Fact 1 imply $\mathbf{E}\big[W^\ell \,|\, F \geq \ell\big] = \mathbf{E}\Big[\sum_{j=1}^k X_j^\ell \,|\, F \geq \ell\Big] \leq (k-1)(2\tau/k)^2$. We can therefore conclude with:

Theorem 3. *If the request sequence is generated by a concentrated adversary,*

$$\frac{\mathbf{E}[LRU(\sigma)]}{\mathbf{E}[OPT(\sigma)]} \leq \frac{8\tau^2}{k} + 2.$$

3.3 Concave adversaries

In this section we provide an analysis of LRU for the case of concave adversaries. We start with the following Lemma:

Lemma 4. *If the request sequence is generated by a concave adversary and $M \geq 4k$,*

$$\mathbf{P}[X_j^\ell = 1 \,|\, F \geq \ell] \leq 2\mathbf{P}[Y_j^\ell = 1 \,|\, F \geq \ell] .$$

Corollary 1. *If the request sequence is generated by a concave adversary and $M \geq 4k$,*

$$\frac{\mathbf{E}\big[W^\ell \,|\, F \geq \ell\big]}{\mathbf{E}[N^\ell \,|\, F \geq \ell]} \leq 2.$$

Corollary 1 and Equation (1) immediately imply the first, following Theorem:

Theorem 4. *If the request sequence is generated by a concave adversary and $M \geq 4k$,*

$$\frac{\mathbf{E}[LRU(\sigma)]}{\mathbf{E}[OPT(\sigma)]} \leq 6.$$

Theorem 4 does not necessarily hold for other marking algorithms, since its proof crucially uses Fact 2. Observe, also, the interesting circumstance that this adversary includes the uniform distribution as a special case. For this particular distribution, the results of Young [22, 24] imply a competitive ratio $\Omega(1)$ as $M = k + o(k)$, that rapidly becomes $\Omega(\log k)$ as $M = k + O(1)$.

3.4 General adversaries

If $D(i) \in \mathcal{G}(\tau) \cup \mathcal{D}_1 \cup \mathcal{D}_2$ for every i, the following is a corollary of the results of Subsections 3.2, 3.3 and 3.1:

Theorem 5. *If the request sequence is generated by a general adversary,*

$$\frac{\mathbf{E}[LRU(\sigma)]}{\mathbf{E}[OPT(\sigma)]} \leq \max\left\{22, \frac{8\tau^2}{k} + 2\right\}.$$

4 A lower bound for FWF

We prove that under very general distributions, the performance of FWF degrades to $\Omega(k)$-competitiveness as locality increases, while the opposite occurs for LRU. This behaviour is in line with practical experience (see e.g. [20, 3]). Together with the results of Section 3, this seems to us one theoretical validation of the great difference observed between the behaviour of FWF and other marking algorithms (in particuar LRU). We prove the following

Theorem 6. *Assume* $0 < \alpha < 1$ *and assume request sequences of length* $n \geq 2(k+1)/\alpha$ *are generated by a diffuse adversary that, for every* $i \leq n$ *and for every* σ *such that* $|\sigma| = i - 1$*, satisfies the following constraint*

$$\mathbf{P}[\Gamma(i) \in \mathcal{R}(\sigma) \,|\, \overline{\Gamma}(\sigma) = \sigma] = 1 - \alpha.$$

Then,

$$\frac{\mathbf{E}[FWF(\sigma)]}{\mathbf{E}[OPT(\sigma)]} > \frac{k}{2((k-1)\alpha + 1)}.$$
$$\frac{\mathbf{E}[LRU(\sigma)]}{\mathbf{E}[OPT(\sigma)]} \leq 2\alpha k + 2.$$

Observe that α measures the amount of locality, i.e. locality increases as α becomes smaller. This theorem thus states that the long term competitive ratio of FWF can be as bad as $\Omega(k)$ as locality increases, whereas under the same circumstances the competitive ratio of LRU tends to become constant. As an example, in the case of a general adversary Theorem 5 implies that this occurs as the standard deviation becomes $O(\sqrt{k})$.

Acknowledgments. The author wishes to thank his colleagues and friends Stefano Leonardi and Alberto Marchetti-Spaccamela for their suggestions and support.

References

1. S. Albers, L. M. Favrholdt, and O. Giel. On paging with locality of reference. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC-02)*, pages 258–267, 2002.
2. L. A. Belady. A study of replacement algorithms for a virtual-storage computer. *IBM Systems Journal*, 5(2):78–101, 1966.
3. A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 1998.
4. A. Borodin, S. Irani, P. Raghavan, and B. Schieber. Competitive paging with locality of reference. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 249–259, 1991.
5. J. Boyar, L. M. Favrholdt, and K. S. Larsen. The relative worst case order ratio applied to paging. In *Proceedings of the 5th Italian Conference on Algorithms and Complexity*, pages 58–69, 2003.

6. J. Boyar, L. M. Favrholdt, and K. S. Larsen. The relative worst order ratio applied to paging. Tech. report ALCOMFT-TR-03-32, Future and Emerging Technologies program under the EU, contract number IST-1999-14186, 2003.
7. P. J. Denning. The working set model for program behaviour. *Communications of the ACM*, 11(5):323–333, 1968.
8. A. Fiat and A. R. Karlin. Randomized and multipointer paging with locality of reference. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on the Theory of Computing*, pages 626–634, 1995.
9. A. Fiat, R. Karp, M. Luby, L. McGeoch, D. Sleator, and N. E. Young. Competitive paging algorithms. *Journal of Algorithms*, 12(4):685–699, 1991.
10. A. Fiat and M. Mendel. Truly online paging with locality of reference. In *38th IEEE Annual Symposium on Foundations of Computer Science*, pages 326–335, 1997.
11. P. A. Franaszek and T. J. Wagner. Some distribution-free aspects of paging performance. *Journal of the ACM*, 21:31–39, 1974.
12. S. Irani, A. R. Karlin, and S. Phillips. Strongly competitive algorithms for paging with locality of reference. *SIAM Journal on Computing*, 25(3):477–497, 1996.
13. A. R. Karlin, S. J. Phillips, and P. Raghavan. Markov paging. In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, pages 208–217, 1992.
14. E. Koutsoupias and C. H. Papadimitriou. Beyond competitive analysis. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 394–400, 1994.
15. M.Chrobak and J. Noga. LRU is better than FIFO. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-98)*, pages 78–81, 1998.
16. S. M. Ross. *Probability Models for Computer Science*. Adademic Press, 2002.
17. D. Sleator and R. E. Tarjan. Amortized Efficiency of List Update and Paging Rules. *Communications of the ACM*, 28:202–208, 1985.
18. Andrew S. Tanenbaum. *Modern Operating Systems*. Prentice Hall, 1992.
19. E. Torng. A unified analysis of paging and caching. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 194–203, 1995.
20. N. Young. The k-server dual and loose competitiveness for paging. *Algorithmica*, 11(6):525–541, 1994.
21. N. E. Young. On-line file caching. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-98)*, pages 82–86, 1998.
22. N. E. Young. On-line paging against adversarially biased random inputs. *Journal of Algorithms*, 37(1):218–235, 2000.
23. Neal Young. *Competitive paging and dual-guided on-line weighted caching and matching algorithms*. PhD thesis, Department of Computer Science, Princeton University, 1991.
24. Neal E. Young. Bounding the diffuse adversary. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 420–425, 1998.