Alcom-FT Technical Report Series
ALCOMFT-TR-02-56

On Paging with Locality of Reference

Susanne Albers[*] Lene M. Favrholdt[†] Oliver Giel[‡]

Abstract

Motivated by the fact that competitive analysis yields too pessimistic results when applied to the paging problem, there has been considerable research interest in refining competitive analysis and in developing alternative models for studying online paging. The goal is to devise models in which theoretical results capture phenomena observed in practice.

In this paper we propose a new, simple model for studying paging with locality of reference. The model is closely related to Denning's working set concept and directly reflects the amount of locality that request sequences exhibit. We demonstrate that our model is reasonable from a practical point of view.

We use the page fault rate to evaluate the quality of paging algorithms, which is the performance measure used in practice. We develop tight or nearly tight bounds on the fault rates achieved by popular paging algorithms such as LRU, FIFO, deterministic Marking strategies and LFD. It shows that LRU is an optimal online algorithm, whereas FIFO and Marking strategies are not optimal in general. We present an experimental study comparing the page fault rates proven in our analyses to the page fault rates observed in practice. This is the first such study for an alternative/refined paging model.

[*]Institut für Informatik, Albert-Ludwigs-Universität, Georges-Köhler-Allee 79, 79110 Freiburg, Germany. E-mail: salbers@informatik.uni-freiburg.de. Work supported by the Deutsche Forschungsgemeinschaft, Project AL 464/3-1, and by the EU, Project APPOL.

[†]Dept. of Mathematics and Computer Science, Odense University, Campusvej 55, DK-5230 Odense M, Denmark. E-mail:lenem@imada.sdu.dk. Supported in part by the Danish Natural Science Research Council (SNF) and in part by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT). Part of this work was done while visiting Universität Dortmund.

[‡]Lehrstuhl Informatik 2, Universität Dortmund, 44221 Dortmund, Germany. E-mail: giel@ls2.cs.uni-dortmund.de.

1 Introduction

Paging is a fundamental and extensively studied problem. Consider a two-level memory system consisting of a small fast memory, that can hold k pages, and a large slow memory. The system must serve a sequence of requests to memory pages. A request can be served if the page to be accessed is in fast memory. If a requested page is not in fast memory, a *page fault* occurs. The missing page must then be loaded into fast memory and, simultaneously, a page must be evicted from fast memory in order to make room for the new page. A paging algorithm decides which page to evict on a fault. This decision must usually be made *online*, i.e., without knowledge of any future requests. The goal is to minimize the number of page faults.

Early work on paging analyzed online algorithms assuming that request sequences are generated by probability distributions, see e.g., [10]. In [15] Sleator and Tarjan introduced competitive analysis and showed that the paging strategies LRU and FIFO achieve an optimal competitive ratio of k. An online algorithm A is c-competitive if, for all request sequences, the number of page faults incurred by A is at most c times the number of faults incurred by an optimal offline algorithm. Practitioners criticized these results because in practice LRU and FIFO achieve performance ratios that are much smaller than k. An experimental study presented by Young [18] shows ratios between 1.2 and 3. It is also known that LRU outperforms FIFO and general deterministic Marking strategies, such as FWF (Flush-When-Full), which are also k-competitive [17]. Thus, competitive analysis does not properly discern between the behavior of different algorithms. The flaw of competitive analysis is that it considers arbitrary request sequences, whereas in practice request sequences have some structure, i.e., they exhibit *locality of reference*.

For this reason there has been considerable research interest in refining competitive analysis and developing alternative models for studying online paging. Young [18] and Borodin et al. [2] initiated this line of research. Young [18] defined the notion of loose competitiveness, where paging algorithms are evaluated for varying fast memory sizes, ignoring sizes where the number of page faults is very small. Borodin et al. [2] introduced the concept of access graphs to model locality of reference. In an access graph G, each node represents a memory page. A request sequence is consistent with G if a request to a page p is followed by a request to a page that is adjacent to p in the graph. Access graphs were also studied in a number of subsequent papers [4, 8, 9, 11]. It was shown that paging algorithms taking into account the underlying access graph can outperform standard paging algorithms and that the competitiveness of LRU is never worse than that of FIFO. In [12] Karlin et al. modeled locality of reference by assuming that request sequences are generated by a Markov chain. They analyzed the page fault rate of paging algorithms and developed an algorithm that achieves an optimal fault rate, for any Markov chain. Torng [17] analyzed the total access time of paging algorithms. He assumes that the service of a request to a page in fast memory costs 1, whereas a fault incurs a penalty of p, $p > 1$. In his model a request sequence exhibits locality of reference if the average length of a subsequence containing requests to m distinct pages is much larger than m. Koutsoupias and Papadimitriou [13] proposed the diffuse adversary model for studying general online algorithms. In this model a request sequence is generated by a probability distribution D that is chosen from a class Δ of distributions known to an algorithm. Koutsoupias and Papadimitriou also introduced a comparative analysis which compares the performance of algorithms from given classes of algorithms.

In this paper we propose a new model for studying paging with locality of reference.

- The model is very simple and closely related to Denning's working set model [6]. It directly reflects the amount of locality exhibited by request sequences. We restrict the class of request sequences from which an adversary may choose a sequence but make no probabilistic assumptions regarding the input.

- We evaluate paging algorithms in terms of their fault rate, the performance measure used by practitioners. We give tight or nearly tight bounds on the fault rates achieved by LRU, FIFO, deterministic Marking strategies and LFD. We show that LRU is an optimal online algorithm in our model but that FIFO and Marking strategies are not optimal in general.
- We have performed an experimental study with request sequences from standard corpora, comparing the fault rates proven in our analyses to the fault rates observed in practice. The gap between the theoretical and observed fault rates is considerably smaller than the corresponding gap in competitive analysis. This is the first time that the theoretical bounds developed in an alternative paging model are compared to the performance observed in practice.

2 The Model

In modeling locality of reference we go back to the working set concept by Denning [6, 7] that is also used in standard text books on operating systems [5, 16] to describe the phenomenon of locality. In practice, a process references during any phase of execution only a relatively small fraction of its pages. The set of pages that a process is currently using is called the *working set*. Determining the working set size in a window of size n at any point in a request sequence, one obtains, for variable n, a function whose general behavior is depicted in Figure 1. The function is increasing and concave. Denning [6] shows that this is in fact a mathematical consequence of the working set model, assuming statistical regularities locally in a request sequence.

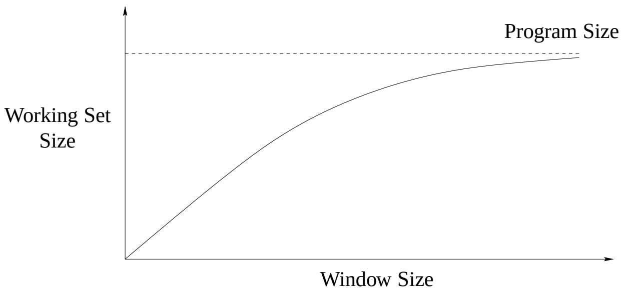


Figure 1: Working set size as a function of the window size.

We will use this simple and natural model to study locality of reference in online paging from a theoretical point of view. We assume that an application is characterized by a concave function f. The application generates request sequences that are *consistent with* f. We will investigate two models. In the *Max-Model* a request sequence is consistent with f if the maximum number of distinct pages referenced in a window of size n is at most $f(n)$, for any $n \in \mathbb{N}$. In the *Average-Model* a request sequence is consistent with f if the average number of distinct pages referenced in a window of size n is at most $f(n)$, for any $n \in \mathbb{N}$. The Max-Model is closely related to the original working set model. On the other hand, the Average-Model permits a larger class of request sequences. It is interesting if an application changes the working set completely at certain times in a request sequence.

In our model the function f characterizes the maximal/average working set size globally in a request sequence, whereas the original working set model considers working set sizes locally. We performed extensive experiments with traces from standard corpora, analyzing maximum/average working set sizes in windows of size n, see Section 7 for details. In all of the cases, the functions have an overall concave shape. Even in very large windows, the number of distinct pages referenced is very small. This demonstrates that the model we propose here is indeed reasonable for studying paging algorithms. What properties do relevant functions f have, apart from being increasing and concave?

Since windows of size 1 contain exactly one page, $f(1) = 1$. If windows of size n contain at most m pages, then a window of size $n+1$ can contain at most $m+1$ pages. Thus, in the Max-Model, f is *surjective* on the integers between 1 and its maximum value, i.e., for all natural numbers m between 1 and $\sup\{f(n) \mid n \in \mathbb{N}\}$, there exists an n with $f(n) = m$. Moreover, $f(n+1) - f(n) \leq 1$.

For a given application, a good approximation of f is easy to determine. One only has to scan a sufficiently long request sequence and compute the maximum/ average number of pages in windows of size n. A function obtained by analyzing real data might not be concave in all intervals. However, this is no problem. Essentially, we can use any concave function f that is an upper bound on the observed data points, e.g., we can take the upper convex hull of the points. We only need that $f(n)$ is an upper bound on the maximum/average number of pages in windows of size n, and $f(n)$ need not even be integral for all n. Therefore, we will work with general functions $f\colon \mathbb{N} \to \mathbb{R}_+$, which will allow us to state concavity in a simple way.

Definition 1. A function $f\colon \mathbb{N} \to \mathbb{R}_+$ is concave* if

i) $f(1) = 1$ and

ii) $\forall n \in \mathbb{N}\colon 0 \leq f(n+2) - f(n+1) \leq f(n+1) - f(n) \leq 1$.

In the Max-Model we additionally require that f be surjective on the integers between 1 and its maximum value.

Both in the Max- and in the Average-Model, given a concave* function f, we will analyze the performance of paging algorithms on request sequences that are consistent with f. Practitioners use the *fault rate* to evaluate the performance of paging algorithms. We will use this measure, too. For a paging algorithm $\mathcal{A}$ and a request sequence σ, let $\mathcal{A}(\sigma)$ be the number of page faults incurred by $\mathcal{A}$ on σ and let $|\sigma|$ be the length of σ. The fault rate of $\mathcal{A}$ on σ is $F_{\mathcal{A}}(\sigma) = \mathcal{A}(\sigma)/|\sigma|$. We are interested in the worst case performance on all sequences that are consistent with f.

Definition 2. The fault rate of a paging algorithm $\mathcal{A}$ with respect to a concave* function f is

$$F_{\mathcal{A}}(f) := \inf\{r \mid \exists n \in \mathbb{N}\colon \forall \sigma, \sigma \text{ consistent with } f, |\sigma| \geq n\colon F_{\mathcal{A}}(\sigma) \leq r\}.$$

Throughout the paper, we will assume that the functions considered are concave*. Moreover, we assume that the functions have maximum values of at least $k+1$, since otherwise the fault rate of any reasonable paging algorithm is 0.

3 Algorithms

We briefly describe the algorithms analyzed in this paper.

- *LRU (Least-Recently-Used)*: On a fault, evict the page whose most recent request was earliest.
- *FIFO (First-In First-Out)*: On a fault, evict the page that has been in fast memory longest.
- *Deterministic Marking strategies*: A request sequence is processed in phases. At the beginning of a phase, all pages are unmarked. Whenever a page is requested, it is marked. On a fault, an arbitrary unmarked page is evicted from fast memory. A phase ends immediately before a fault when there are k marked pages in fast memory.
- *LFD (Longest-Forward-Distance)*: Evict the page whose next request is farthest in the future.

LFD is an optimal offline algorithm, i.e., on any request sequence it achieves the minimum number of page faults [1].

4 Results

Both for the Max- and the Average-Model we develop tight or nearly tight bounds on the fault rates achieved by popular paging algorithms such as LRU, FIFO, deterministic Marking strategies and LFD. The first three strategies work online. On the other hand, LFD (Longest-Forward-Distance) is an offline algorithm that cannot be applied in practice. However, since LFD is an optimal offline algorithm, it is interesting to analyze its fault rate.

In Section 5, we investigate the Max-Model. We first prove that no deterministic online paging algorithm can have a fault rate smaller than $\frac{k-1}{f^{-1}(k+1)-2}$, where f^{-1} is the inverse function of f, formally defined in Section 5. We then show that LRU is an optimal online algorithm in our model. It achieves a fault rate that is exactly equal to the lower bound. LRU is a special Marking strategy. We show, however, that general deterministic Marking strategies are not as good as LRU. Any Marking strategy has a fault rate of at most $\frac{k}{f^{-1}(k+1)-1}$. This bound cannot be improved in general; there are Marking strategies that have a fault rate exactly equal to that bound. We analyze FIFO and prove that it has a fault rate of at least $\frac{k-1/k}{f^{-1}(k+1)-1}$ and at most $\frac{k}{f^{-1}(k+1)-1}$. The gap between the lower bound for FIFO and the best possible fault rate achieved by LRU is small. However, in our experiments the difference in the fault rates observed for LRU and FIFO is also small, see Section 7. We finally study LFD and show that its fault rate depends on the total number M of pages that may be requested. We show that LFD has a fault rate of at least $\max\{\frac{m}{f^{-1}(k+m)-1}\}$, where the maximum is taken over all positive integers m with $m + k \leq M$. We prove an upper bound that is at most a factor of 2 away from this lower bound.

In Section 6, we study the Average-Model. We prove that every deterministic online paging algorithm has a fault rate of at least $\frac{f(k+1)-1}{k}$. We show that in the Average-Model, both LRU and FIFO are optimal and achieve a fault rate equal to the lower bound. On the other hand, general Marking strategies are considerably worse. A general Marking strategy has a page fault rate of essentially $\frac{4}{3}\frac{f(k+1)}{k}$. If k is even, the exact bound is $\frac{4k}{3k+2}\frac{f(k+1)}{k}$. If k is odd, then there is an additive $-1/k$ in the denominator of the first term. The bounds cannot be improved in general. There are Marking strategies and functions f for which the bounds are tight. We also develop tight bounds for LFD. The fault rates depend again on the total number M of pages that may be requested. If k is odd, then the exact fault rate is $\frac{4M-4k}{4M-k-3}\frac{f(k+1)}{k+1}$. If k is even, there is an additive $-1/(k+1)$ in the denominator of the first term. If M is approximatly k, LFD has page fault rate close to 0, as expected. If M is large compared to k, the fault rate is close to $\frac{f(k+1)}{k+1}$.

Table 1 summarizes the results presented in Section 5 and Section 6. For simplicity, we sometimes only give the essential terms here, yielding slightly larger bounds. Upper and lower bounds on the fault rates of the considered algorithms are indicated by "$\leq$" and "$\geq$", respectively, and "$=$" and "$\approx$" indicate tight bounds.

In Section 7, we present the experimental study mentioned already a few times in this text. We first demonstrate that our models for quantifying locality is indeed reasonable from a practical point of view and then compare the fault rates developed in our models to the fault rates observed in practice. For the Max-Model, the results are quite good. The gap between the theoretical and observed bounds is considerably smaller than the corresponding gap in competitive analysis, unless the size of the fast memory is extremely small. As the size of the fast memory increases, the gap decreases and is very small for large fast memories. For the Average-Model, the results are not as good. Here we still have a considerable gap between the theoretical and observed fault rates. Our explanation for this phenomenon is as follows. The Average-Model permits a larger class of request sequences than the Max-Model. This larger class may contain request sequences that cause high fault rates

	Max-Model	Average-Model
Online	$\geq \frac{k-1}{f^{-1}(k+1)-2}$	$\geq \frac{f(k+1)-1}{k}$
LRU	$= \frac{k-1}{f^{-1}(k+1)-2}$	$= \frac{f(k+1)-1}{k}$
FIFO	$\leq \frac{k}{f^{-1}(k+1)-1}, \quad \geq \frac{k-1/k}{f^{-1}(k+1)-1}$	$= \frac{f(k+1)-1}{k}$
Marking	$\leq \frac{k}{f^{-1}(k+1)-1}$	$\leq \frac{4}{3}\frac{f(k)}{k}$
LFD	$\geq \max\limits_{\substack{m\in\mathbb{N}\\ k+m\leq M}} \left\{\frac{m}{f^{-1}(k+m)-1}\right\}, \leq 2 \max\limits_{\substack{1\leq m\leq k\\ k+m\leq M}} \left\{\frac{m+1}{f^{-1}(k+m)}\right\}$	$\approx \frac{4(M-k)}{4M-k}\frac{f(k+1)}{k+1}$

Table 1: Fault rates of all algorithms considered in this work.

in the mathematical analyses but typically do not occur in practice. We conclude that while the Average-Model is interesting from a mathematical point of view, the Max-Model seems to model more accurately the request sequences that occur in practice.

5 Paging in the Max-Model

We study the Max-Model. Given a concave* function f, $f(n)$ is an upper bound on the maximum number of distinct pages encountered in any n consecutive requests of a request sequence. In this section we will assume that $f(2) = 2$, because $f(2) < 2$ only permits request sequences referencing a single page for which the page fault rate of any reasonable algorithm is 0. Furthermore, we consider the case $k \geq 2$. If $k = 1$ an adversary can easily cause a fault rate of 1 for any paging algorithm because $f(2) = 2$. For the analyses of the fault rates we need to define the inverse function of f. Let $M = \sup\{\lfloor f(n)\rfloor \mid n \in \mathbb{N}\}$. Define $f^{-1}\colon \{m \in \mathbb{N} \mid m \leq M\} \to \mathbb{N}$ by

$$f^{-1}(m) := \min\{n \in \mathbb{N} \mid f(n) \geq m\}.$$

Thus, $f^{-1}(m)$ is the smallest possible size of a window containing m distinct pages. The following proposition will be crucial in our analyses.

Proposition 1. *f^{-1} is a strictly increasing function satisfying*

$$f^{-1}(m) - f^{-1}(m-1) \geq f^{-1}(m-1) - f^{-1}(m-2), \quad \textit{for all } 3 \leq m \leq M.$$

Proof. We prove the stated inequality. Since $f^{-1}(2) - f^{-1}(1) = 1$, this immediately implies that f^{-1} is strictly increasing. The function f is concave*, i.e., $f(n+1) - f(n) \geq f(n+2) - f(n+1)$. Thus, for any $n, n', k \in \mathbb{N}$ such that $n' \geq n + k$,

$$\begin{aligned} f(n+k) - f(n) = \sum_{i=n}^{n+k-1} \big(f(i+1) - f(i)\big) &\geq \sum_{j=n'-k}^{n'-1} \big(f(j+1) - f(j)\big) \\ &= f(n') - f(n'-k). \end{aligned} \tag{1}$$

The inequality in the middle holds because, the number of terms in both sums is the same and, for each index i on the left, there is an index $j \geq i$ on the right.

Let $n = f^{-1}(m-2)$ and $n' = f^{-1}(m)$. Since f is surjective we have $f(n) = m-2$ and $f(n') = m$. Set $k = \lfloor \frac{n'-n}{2} \rfloor$. Using (1) we obtain

$$\frac{f(n+k) + f(n'-k)}{2} \geq \frac{f(n) + f(n')}{2} = m - 1. \tag{2}$$

If $n' - n$ is even, then $f(n+k) = f(n'-k) = m-1$ and $f^{-1}(m-1) \leq n+k$. The desired inequality holds because $n+k = n'-k$ and

$$\begin{aligned} f^{-1}(m) - f^{-1}(m-1) &\geq n' - (n+k) = n' - k - n \\ &\geq f^{-1}(m-1) - f^{-1}(m-2). \end{aligned} \tag{3}$$

If $n'-n$ is odd, then $n+k+1 = n'-k$. Inequality (2) ensures that $f(n+k) \geq m-1$. Otherwise, if $f(n+k) < m-1$, we had $f(n+k+1) = f(n'-k) > m-1$ and f would not be surjective. Thus, $f^{-1}(m-1) \leq n+k$ and we can again derive (3) because $n+k \leq n'-k$. □

We first develop a lower bound on the fault rate that can be achieved by any deterministic online paging algorithm and then show that LRU is optimal.

Theorem 1. *Let $\mathcal{A}$ be any deterministic online paging algorithm. Then*

$$F_{\mathcal{A}}(f) \geq \frac{k-1}{f^{-1}(k+1) - 2} \ .$$

Proof. We construct a family of request sequences σ_n, where the length n of a sequence can be made arbitrarily large, such that $\mathcal{A}$'s fault rate on any of the sequences is at least the desired bound. We need $k+1$ distinct pages $p_1, \ldots, p_{k+1}$. A request sequence is constructed in *phases*, each of which has a length of $f^{-1}(k+1) - 2$ and is composed of $k-1$ *blocks*. A block is a subsequence of requests, all to the page that was not in $\mathcal{A}$'s fast memory at the end of the previous block. Thus, $\mathcal{A}$ has a cost of 1 in each block and a cost of $k-1$ in each phase. In each phase, block j, $1 \leq j \leq k-1$, starts with request $f^{-1}(j+1) - 1$. Note that the partitioning of the phases into blocks is well-defined, since $f(2) = 2$. Thus, the first block of a phase starts with the first request of the phase. Within a phase, block j, $1 \leq j \leq k-1$, has a length of $\big(f^{-1}(j+2) - 1\big) - \big(f^{-1}(j+1) - 1\big) = f^{-1}(j+2) - f^{-1}(j+1)$. By Proposition 1, f^{-1} is strictly increasing. Thus, the blocks are non-empty and the constructed sequence is well-defined. Also, within a phase the block lengths are non-decreasing.

It remains to show that the request sequence is consistent with f. To this end it suffices to show that any subsequence with j distinct pages has a length of at least $f^{-1}(j)$. For $1 \leq j \leq 2$, there is nothing to show because $f^{-1}(j) = j$ in this case. The most interesting range of j is $3 \leq j \leq k$. Any subsequence with j distinct pages must (partially) cover at least j consecutive blocks. Since the blocks are homogenous with respect to the requested page, a subsequence of minimal length with j distinct pages only contains the last request of the first block partially covered and, analogously, only the first request of the last block partially covered. Extending the subsequence further into the first or last block, we do not gain any additional pages but only increase the length of the subsequence. As stated above, the block lengths in a phase are non-decreasing. Thus, a subsequence with j distinct pages of minimal length fully covers the first $j-2$ blocks of a phase and includes the last request of the previous phase as well as the first request of block $j-1$. The length is $\big(f^{-1}(j) - 1\big) + 1 = f^{-1}(j)$.

We finally have to consider $j = k+1$. A subsequence with $k+1$ distinct pages must partially include at least $k+1$ blocks and has a length of at least $\big(f^{-1}(k+1) - 2\big) + 2 = f^{-1}(k+1)$. □

Theorem 2. *The fault rate of LRU is* $F_{LRU}(f) \le \dfrac{k-1}{f^{-1}(k+1)-2}$.

Proof. Let σ be an arbitrary request sequence consistent with f. We assume that the initial fast memory is empty. We partition the request sequence into phases such that each phase contains exactly $k-1$ faults made by LRU (except for possibly the last phase) and starts with a fault. In general, the ith phase, $i \ge 2$, starts with the $((i-1)(k-1)+1)$st fault and ends immediately before the $(i(k-1)+1)$st fault. The last phase might be incomplete. LRU incurs a cost of at most $k-1$ per phase. We show that each phase, except for possibly the first and the last one, has a length of at least $f^{-1}(k+1)-2$. Consider an arbitrary phase P different from the first and the last phase. We argue that the subsequence of σ starting at the last request before P and ending at the first request after P (including that request) contains $k+1$ distinct pages. This implies that P has a length of at least $f^{-1}(k+1)-2$. Let x be the page referenced by the last request before P. Phase P and the first request after P include k page faults. If these page faults are on distinct pages different from x, then we are done. If one of the faults is on x, then x must have been evicted in P at some fault to a page y. At that time x was the least recently requested page in fast memory and hence we have identified $k+1$ distinct pages in our subsequence. The same argument applies to the case that LRU faults twice on requests to some page z, $z \ne x$. To conclude, σ consists of at most

$$1 + \left\lceil \frac{|\sigma| - \ell}{f^{-1}(k+1)-2} \right\rceil \;\le\; \frac{|\sigma|}{f^{-1}(k+1)-2} + 2$$

phases, where ℓ denotes the length of the first phase. In each phase LRU has at most $k-1$ faults. Thus, the fault rate on σ is bounded by

$$\frac{k-1}{f^{-1}(k+1)-2} + \frac{2k-2}{|\sigma|} \;,$$

where the last term gets arbitrarily small for increasing $|\sigma|$. $\square$

LRU is a special Marking strategy. We show, however, that Marking algorithms in general are not as good as LRU, i.e., there are Marking algorithms that have a higher fault rate. In the following we first give an upper bound and then provide a matching lower bound.

Theorem 3. *The fault rate of any Marking algorithm* $\mathcal{M}$ *is*

$$F_{\mathcal{M}}(f) \le \frac{k}{f^{-1}(k+1)-1} \;.$$

Proof. A Marking algorithm $\mathcal{M}$ partitions a request sequence σ into phases consisting of requests to k distinct pages (except for possibly the last one) such that it incurs a fault on the first request of each phase. Any subsequence that starts at the beginning of the phase and ends immediately after the first request of the next phase has length $f^{-1}(k+1)$ because the k pages requested in the phase are all different from the first page requested in the next phase. Thus, all but the last phase have a length of at least $f^{-1}(k+1)-1$ each. The request sequence consists of at most

$$\left\lceil \frac{|\sigma|}{f^{-1}(k+1)-1} \right\rceil \;\le\; \frac{|\sigma|}{f^{-1}(k+1)-1} + 1$$

phases, each causing at most k faults. Thus, the fault rate on σ is bounded by

$$\frac{k}{|\sigma|}\left(\frac{|\sigma|}{f^{-1}(k+1)-1} + 1\right) \;\le\; \frac{k}{f^{-1}(k+1)-1} + \frac{k}{|\sigma|} \;.$$

$\square$

The next theorem implies that the upper bound of Theorem 3 cannot be improved in general.

Theorem 4. *There are Marking strategies $\mathcal{M}^*$ whose fault rates are*

$$F_{\mathcal{M}^*}(f) \geq \frac{k}{f^{-1}(k+1)-1} \quad .$$

Proof. We simultaneously describe the family of request sequences σ_n and the behavior of the Marking algorithms M^*. As usual we need a set of $k+1$ pages $p_1, \ldots, p_{k+1}$. A request sequence consists of the *phases* constructed by the given Marking algorithm. Each phase is composed of k *blocks*, where a block is a subsequence of requests to the same page. Within a phase, block j has a length of $f^{-1}(j+1) - f^{-1}(j)$, for $1 \leq j \leq k$. Proposition 1 ensures that the block lengths are well-defined, i.e., they are non-zero, and non-decreasing in a phase. The total length of a phase is $f^{-1}(k+1) - 1$.

We assume that the initial fast memory is empty. In the first phase, the jth block consists of requests to p_j, $1 \leq j \leq k$. Suppose that we have already constructed i phases such that each phase contains exactly k distinct pages. We show how to construct the $(i+1)$st phase. The first block of phase $i+1$ consists of $f^{-1}(2) - f^{-1}(1) = 2-1$ request to the unique page that it unmarked at the end of phase i. The Marking algorithm M^* has a fault on this request. We assume that M^* evicts the page that was requested in the last block of phase i. This assumption is feasible because at the beginning of a phase all pages are unmarked and a Marking strategy, on a fault, evicts an arbitrary unmarked page. Each of the next $k-1$ blocks of the phase references the page that is not in the fast memory of $\mathcal{M}^*$ at the beginning of that block. Thus, $\mathcal{M}^*$ has a total of k faults in a phase, which gives the desired fault rate. The pages requested in the k blocks of a phase are distinct. By construction, the page requested in the second block of a phase is equal to the page requested in the last block of the previous phase.

It remains to prove that the request sequence is consistent with f. We show that any subsequence with j distinct pages has a length of at least $f^{-1}(j)$, $1 \leq j \leq k+1$. For $j \in \{1,2\}$, there is nothing to show because $f^{-1}(j) = j$ for these two values. For any j with $3 \leq j \leq k+1$, a subsequence with j distinct pages must partially cover at least j consecutive blocks because blocks are homogenous with respect to the requested page. The block lengths are non-decreasing in a phase. Thus, if $3 \leq j \leq k$, a subsequence with j distinct pages of minimal length starts at the beginning of a phase and ends after the first request of block j. The length is exactly $f^{-1}(j)$. The final case $j = k+1$ needs some extra arguments. A subsequence with $k+1$ distinct pages must contain requests from two consecutive phases. If the subsequence fully covers some phase i, then we are done because a phase has length $f^{-1}(k+1) - 1$ and one additional request must be covered. Otherwise the phase partially covers two consecutive phases i and $i+1$. In this case the subsequence must partially cover at least $k+2$ blocks because the page in the kth block of phase i is the same as the second block of phase $i+1$. Since the length of k consecutive blocks is exactly equal to the length of a phase, the subsequence has length at least $\big(f^{-1}(k+1) - 1\big) + 1 = f^{-1}(k+1)$. □

In the following we show that FIFO is not an optimal online algorithm in our model. We first develop a lower bound on FIFO's fault rate and then present a nearly matching upper bound.

Theorem 5. *If $f^{-1}(4) - f^{-1}(3) > f^{-1}(3) - f^{-1}(2)$, then*

$$F_{FIFO}(f) \geq \frac{k - 1/k}{f^{-1}(k+1) - 1} \quad .$$

Straightforward algebraic manipulations show that the fault rate of FIFO given in the last theorem is in fact larger than that of LRU (details omitted here). The condition on f means that there must be some locality in windows of size 5, i.e., $f(5) \leq 4$. We can relax the constraint such that there must be some locality in the request sequence, i.e., $f^{-1}(m) - f^{-1}(m-1) > f^{-1}(m-1) - f^{-1}(m-2)$ for some $m \geq 3$, but then our lower bound becomes slightly weaker.

Proof of Theorem 5. Let $p_0, \ldots, p_k$ be $k+1$ distinct pages. We construct a family of request sequences σ_n. A request sequence consists of an initial request to p_k followed by a sequence of *phases*, each composed of $k-1$ *blocks*. The blocks are not homogeneous; each block consists of one request to some page p_i, $0 \leq i \leq k-1$, followed by one or more requests to p_k, depending on the length of the block. In the sequence of blocks, the pages $p_0, \ldots, p_{k-1}$ are requested in cyclic order, i.e., in the jth block in the request sequence the first request is made to $p_{(j-1) \bmod k}$. The block lengths are as follows. In any phase, the first block has length $f^{-1}(3) - f^{-1}(2) + 1$ and the jth block has a length of $f^{-1}(j+2) - f^{-1}(j+1)$, for $j = 2, \ldots, k-1$. By Proposition 1 and the condition on f, the block lengths are non-decreasing, the first block having a length of $f^{-1}(3) - f^{-1}(2) + 1 \geq 3 - 2 + 1 = 2$. Thus, each block contains at least one request to p_k. The total length of a phase is $f^{-1}(k+1) - f^{-1}(2) + 1 = f^{-1}(k+1) - 1$. In the rest of the proof we will argue that the constructed request sequence is indeed consistent with f and that in any k consecutive phases, which we call a *super phase*, FIFO incurs $(k-1)(k+1)$ faults. This gives a fault rate of

$$\frac{(k-1)(k+1)}{k(f^{-1}(k+1)-1)} = \frac{k-1/k}{f^{-1}(k+1)-1}$$

as desired.

We first prove consistency with f by arguing that any subsequence with j distinct pages has a length of $f^{-1}(j)$. For $j \in \{1, 2\}$ there is nothing to show. Consider a j with $3 \leq j \leq k$. Any subsequence with j distinct pages must span more than a block because a block contains only two distinct pages. A subsequence of minimal length does not start with a prefix of requests to p_k because that page is contained in the next block anyway. Thus, it starts at the beginning of a block and extends at least beyond the first request of the $(j-2)$nd following block. Since block lengths are non-decreasing in a phase, a subsequence with j distinct pages has a length of at least $f^{-1}(j) + 1 - f^{-1}(2) + 1 = f^{-1}(j)$. Finally, a subsequence with $k+1$ distinct pages must span more than a phase and hence its length is at least $f^{-1}(k+1)$.

We now analyze the number of faults made by FIFO in a super phase. Assume that the initial fast memory is empty. FIFO first misses on p_k and then on $p_0, \ldots, p_{k-1}$, which are requested in the next k blocks. On k consecutive faults, FIFO never misses twice on the same page. Thus, the fault sequence is $p_k, p_0, \ldots, p_{k-1}$, which repeats in cyclic order. We show inductively that FIFO misses on the first request of each block. This clearly holds for the first k blocks. Suppose that FIFO misses on the first request of block j, $j \geq k$. If page p_i with $i < k-1$ is requested, then p_{i+1} is evicted, which is referenced in block $j+1$. If p_{k-1} is requested, then p_k is evicted, which is referenced in the same block. The fault on p_k causes an eviction of p_0, requested in the next block. Our proof also shows that FIFO has two page faults on any block with a request to p_{k-1}. In any k consecutive phases, $k-1$ of these contain such a block. Thus, in any super phase the total number of page faults is $k(k-1) + k - 1 = (k+1)(k-1)$. □

We complement our lower bound by giving a nearly matching upper bound.

Theorem 6. *The fault rate of FIFO is* $F_{FIFO}(f) \leq \dfrac{k}{f^{-1}(k+1)-1}$.

Proof. On any $k+1$ consecutive faults in a request sequence σ, FIFO never faults twice on the same page. Partition σ into phases such that each phase contains exactly k faults made by FIFO and starts with a fault. Consider a subsequence that spans one full phase and includes the first request of the next phase. The subsequence covers $k+1$ faults, i.e., $k+1$ distinct pages. Hence, its length is at least $f^{-1}(k+1)$, and the phase length is only 1 smaller $\square$

We next give bounds on the fault rate of LFD.

Theorem 7. *The fault rate of LFD is* $F_{LFD}(f) \geq \max_{\substack{m \in \mathbb{N} \\ k+m \leq M}} \left\{ \frac{m}{f^{-1}(k+m)-1} \right\}$.

Proof. Fix an $m \in \mathbb{N}$ and $N = k+m$ pages $p_0, \ldots, p_{N-1}$. We construct a family of request sequences in phases, where each phase has a length of $f^{-1}(N)-1$. Each phase is composed of $N-1$ blocks, the jth block in a phase having a length of $f^{-1}(j+1) - f^{-1}(j)$, for $j = 1, \ldots, N-1$. In the overall sequence, the pages $p_0, \ldots, p_{N-1}$ are requested in cyclic order, i.e., the jth block consists of requests to page $p_{(j-1) \bmod N}$, for any positive integer j. The page referenced in the last block of a phase is not requested in the following phase but resides in LFD's fast memory at the end of the phase. Thus, among the $N-1$ pages requested in the next phase, only $k-1$ of these can be in LFD's fast memory at the beginning of the phase. Hence, LFD incurs at least $(N-1)-(k-1) = N-k$ faults in a phase. This gives the desired fault rate. As in the previous lower bound proofs we can show that any subsequence with j distinct pages has a length of at least $f^{-1}(j)$, which yields consistency of the constructed request sequence with f. $\square$

We prove an upper bound on LFD's fault rate that is essentially a factor of 2 away from the lower bound. To prove this upper bound we need the following technical proposition.

Proposition 2. *For any* $m_1, \ldots, m_n \in \mathbb{N}$, $\sum_{\ell=1}^{n} f^{-1}(m_\ell) \geq n \cdot f^{-1}\left(\left\lfloor \sum_{\ell=1}^{n} \frac{m_\ell}{n} \right\rfloor \right)$.

Proof. Proposition 1 implies that

$$f^{-1}(m) + f^{-1}(m') \geq f^{-1}(m+1) + f^{-1}(m'-1) \tag{4}$$

for all $m, m' \in \mathbb{N}$ with $m' - m \geq 2$. We now manipulate the sum $\sum_{l=1}^{n} f^{-1}(m_l)$ as follows. At any time we keep a sequence of n terms $f^{-1}(\widetilde{m}_1), \ldots, f^{-1}(\widetilde{m}_n)$, where the arguments $\widetilde{m}_l$ are natural numbers. Initially, $\widetilde{m}_l = m_l$, for $l = 1, \ldots, n$. At any time let $m = \min\{\widetilde{m}_l \mid l = 1, \ldots, n\}$ and $m' = \max\{\widetilde{m}_l \mid l = 1, \ldots, n\}$. While $m' - m \geq 2$, replace two terms $f^{-1}(m)$ and $f^{-1}(m')$ by $f^{-1}(m+1)$ and $f^{-1}(m'-1)$. By (4), this cannot increase the total sum of the terms. When the process terminates, each $\widetilde{m}_l$ is either $\overline{m}$ or $\overline{m}+1$, where $\overline{m} = \lfloor \sum_{l=1}^{n} m_l / n \rfloor$. The proposition then follows because $f^{-1}(\overline{m}+1) \geq f^{-1}(\overline{m})$. $\square$

Theorem 8. *The fault rate of LFD is* $F_{LFD}(f) \leq 2 \max_{\substack{1 \leq m \leq k \\ k+m \leq M}} \left\{ \frac{m+1}{f^{-1}(k+m)} \right\}$.

Proof. Partition a given request sequence into phases such that each phase contains exactly k distinct pages (except for possibly the last phase) and the k pages are all different from the first page requested in the next phase. Suppose that the partitioning consists of p phases $P_1, \ldots, P_p$, such that. For any phase i, let m_i be the number of new pages, i.e., pages referenced in phase i that were not referenced in phase $i-1$. We assume that LFD initially starts with an empty fast memory and set $m_1 = k$. Consider an offline strategy that performs page swaps without evicting pages that are referenced in the phase. The number of page faults made by this algorithm in any phase i is m_i. Since LFD is an optimal offline algorithm, the total number of page faults made by LFD cannot be larger and is bounded by $k + \sum_{i=2}^{p} m_i = k + (p-1)\overline{m}$, where $\overline{m} = \frac{1}{p-1}\sum_{i=2}^{p} m_i$. Any two consecutive phases $i-1$ and i contain $k+m_i$ distinct pages and thus have a length of $|P_{i-1}| + |P_i| \geq f^{-1}(k+m_i)$. The total length of σ is

$$\begin{aligned} |\sigma| \;&=\; \sum_{i=1}^{p} |P_i| \;=\; \frac{1}{2}\sum_{i=2}^{p}(|P_{i-1}| + |P_i|) + \frac{1}{2}(|P_1| + |P_p|) \\ &>\; \frac{1}{2}\sum_{i=2}^{p} f^{-1}(k+m_i). \end{aligned}$$

By Proposition 2, $|\sigma| > \frac{1}{2}(p-1)f^{-1}(k + \lfloor\overline{m}\rfloor)$. LFD's fault rate on σ is

$$F_{\mathrm{LFD}}(\sigma) \leq \frac{k+(p-1)\overline{m}}{|\sigma|} \leq \frac{2\overline{m}}{f^{-1}(k+\lfloor\overline{m}\rfloor)} + \frac{2k}{|\sigma|} \;\; .$$

The second term in the sum becomes arbitrarily small for increasing $|\sigma|$. Thus, LFD's fault rate is

$$F_{\mathrm{LFD}}(f) \leq 2 \max_{\substack{1\leq m\leq k \\ k+m\leq M}} \left\{ \frac{m+1}{f^{-1}(k+m)} \right\} \; .$$

$\square$

6 Paging in the Average-Model

We now turn to the Average-Model. We shall need some additional notation. For any sequence σ of page requests, $\sigma[i]$ denotes the ith request r in σ as well as the page requested by r, $1 \leq i \leq |\sigma|$. For $1 \leq i \leq |\sigma|-\ell+1$, let $\sigma_\ell[i]$ be the subsequence $\langle \sigma[i], \sigma[i+1], \ldots, \sigma[i+\ell-1] \rangle$. Subsequences of σ are also called *windows*. Let $N_\ell(i)$ be the number of distinct pages in $\sigma_\ell[i]$, and let $N_\ell = \sum_{i=1}^{|\sigma|-\ell+1} N_\ell(i)$. Let Av($\ell$) be the average number of distinct pages in windows of length ℓ, i.e., Av(ℓ) $= \frac{N_\ell}{|\sigma|-\ell+1}$. Thus, a sequence σ consistent with a given concave* function f has Av(ℓ) $\leq f(\ell)$, $1 \leq \ell \leq |\sigma|$.

6.1 A Lower Bound for Deterministic Algorithms

In this section we will prove a lower bound of $\frac{f(k+1)-1}{k}$ on the fault rate of any deterministic paging algorithm $\mathcal{A}$ with respect to any concave* function f. We will build sequences consisting of two parts. Each sequence has a prefix on which $\mathcal{A}$ faults on each request. To ensure that the sequences are consistent with f a suffix consisting of requests to only one page is added. As a beginning, consider the sequence $\sigma(n,m) = \langle p_1, p_2, p_3, \ldots, p_k, p_{k+1}\rangle^n \langle p_1 \rangle^m$ consisting of requests to $k+1$ distinct pages, where n is a large integer greater than or equal to $k+2$ and m is an integer greater than or equal to $k+1$. For convenience, we usually omit n and m and refer to the sequence as σ. For $1 \leq \ell \leq n(k+1)$, let $r(\ell) = (k+1)n - \ell + 1$. The window $\sigma_\ell[r(\ell)]$ is the rightmost window completely contained in the prefix $\langle p_1, p_2, p_3, \ldots, p_k, p_{k+1}\rangle^n$. To determine the minimum length m of the suffix ensuring that $\sigma(n,m)$ is consistent with a given concave* function f, we shall need the following upper bound on the average number of distinct pages in windows of length ℓ, $1 \leq \ell \leq |\sigma|$.

Lemma 1. *For any $\ell' \geq k+2$, let $A(\ell)$ be defined as*

$$A(\ell) = \begin{cases} 1 + \Delta_1(\ell - 1), & 1 \leq \ell \leq k+1, \\ (1 + \Delta_1 k) + \Delta_2(\ell - (k+1)), & k+1 \leq \ell \leq \ell', \\ k+1, & \ell \geq \ell', \end{cases}$$

$\Delta_1 = 1 - \dfrac{m-k}{(k+1)n+m}$ *and* $\Delta_2 = \dfrac{(k+1) - (1+\Delta_1 k)}{\ell' - (k+1)}$ *(see Figure 2).*

If $m = qn$, for some constant $q > 0$, there exists an $n_0 \in \mathbb{N}$ such that, for $n \geq n_0$, $1 \leq \ell \leq |\sigma|$, $\mathrm{Av}(\ell) \leq A(\ell)$.

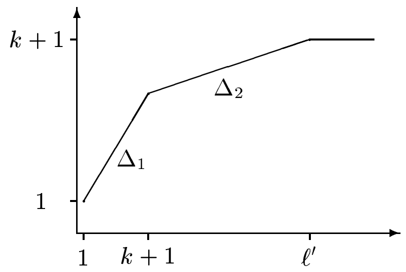


Figure 2: $A(\ell)$, an upper bound on $\mathrm{Av}(\ell)$.

Proof. We have

$$\begin{aligned} \mathrm{Av}(\ell+1) - \mathrm{Av}(\ell) &= \frac{N_{\ell+1}}{(k+1)n+m-\ell} - \frac{N_\ell}{(k+1)n+m-\ell+1} \\ &= \frac{N_{\ell+1} - N_\ell + \mathrm{Av}(\ell)}{(k+1)n+m-\ell}, \end{aligned} \tag{5}$$

$$\begin{aligned} N_{\ell+1} - N_\ell &= \sum_{i=1}^{|\sigma|-\ell} N_{\ell+1}(i) - \sum_{i=1}^{|\sigma|-\ell+1} N_\ell(i) \\ &= \sum_{i=1}^{|\sigma|-\ell} \Big(N_{\ell+1}(i) - N_\ell(i)\Big) - N_\ell(|\sigma| - \ell + 1). \end{aligned} \tag{6}$$

The rest of the proof is divided into three cases, according to the three linear parts of $A(\ell)$.

Case $1 \leq \ell \leq k+1$. Obviously, $\mathrm{Av}(1) = 1 = A(1)$. It remains to prove $\mathrm{Av}(\ell+1) - \mathrm{Av}(\ell) \leq \Delta_1$, for $1 \leq \ell \leq k$. Thus, assume now that $1 \leq \ell \leq k$. Then no window of size ℓ contains all $k+1$ distinct pages.

For $1 \leq i \leq r(\ell)$, $N_{\ell+1}(i) - N_\ell(i) = 1$. For $i \geq r(\ell)+1$, $N_{\ell+1}(i) = N_\ell(i)$, since $\sigma_\ell[i]$ already contains the page p_1. The boundary case is depicted in Figure 3.

Thus, $\sum_{i=1}^{|\sigma|-\ell} \big(N_{\ell+1}(i) - N_\ell(i)\big) = r(\ell)$. Since $\ell \leq k \leq m$, the rightmost window of length ℓ is completely contained in the suffix, so $N_\ell(|\sigma| - \ell + 1) = 1$. Therefore, by (6), $N_{\ell+1} - N_\ell = r(\ell) - 1 =$

$$\langle \underbrace{p_1,\ldots,p_{k+1},\ldots\ldots\ldots\ldots,p_1,\ldots \overbrace{\boxed{\ldots,p_{k+1},}}^{\ell\leq k}}_{(k+1)n}\underbrace{\boxed{p_1,}\ldots,p_1}_{m}\rangle$$

Figure 3: When $\sigma_\ell[r(\ell)]$ is extended to $\sigma_{\ell+1}[r(\ell)]$, p_1 is included in the window.

$(k+1)n-\ell$. Now, by (5) and $\text{Av}(\ell) \leq \text{Av}(k) \leq k$,

$$\begin{aligned}\text{Av}(\ell+1)-\text{Av}(\ell) \leq \frac{(k+1)n-\ell+k}{(k+1)n-\ell+m} &= 1-\frac{m-k}{(k+1)n-\ell+m}\\ &< 1-\frac{m-k}{(k+1)n+m} = \Delta_1.\end{aligned}$$

Case $k+1 \leq \ell \leq \ell'$. It follows from the previous case that $\text{Av}(k+1) \leq 1+\Delta_1 k$. Thus, it suffices to show $\text{Av}(\ell+1)-\text{Av}(\ell) \leq \Delta_2$, for $k+1 \leq \ell \leq \ell'-1$. Observe that for all i, $1 \leq i \leq r(\ell)$, it holds that $N_\ell(i) = k+1 = N_{\ell+1}(i)$ because $\ell \geq k+1$. Also for all $i > r(\ell)$, $N_\ell(i) = N_{\ell+1}(i)$ because p_1 is already included in $N_\ell(i)$. So $N_{\ell+1}(i) - N_\ell(i) = 0$, for $1 \leq i \leq |\sigma|-\ell$. Thus, by (6), $N_{\ell+1} - N_\ell = 0 - N_\ell(|\sigma|-\ell+1) \leq -1$. Now, by (5),

$$\text{Av}(\ell+1)-\text{Av}(\ell) \leq \frac{-1+(k+1)}{(k+1)n+m-\ell} \leq \frac{k}{(k+1)n+m-\ell'}.$$

For any fixed ℓ' and k, $\text{Av}(\ell+1)-\text{Av}(\ell) \in O\left(\frac{1}{n}\right)$, and

$$\Delta_2 = \frac{k(m-k)}{((k+1)n+m)(\ell'-(k+1))} = \frac{kqn-k^2}{(k+1+q)n(\ell'-(k+1))} = \frac{an-b}{cn},$$

$a, b, c \in O(1)$. Thus, $\Delta_2 \in O(1)$. Therefore, there exists an $n_0 \in \mathbb{N}$ such that $\text{Av}(\ell+1)-\text{Av}(\ell) \leq \Delta_2$, for $n \geq n_0$.

Case $\ell \geq \ell'$. Since there are only $k+1$ distinct pages, it follows that $\text{Av}(\ell) \leq k+1$, for all ℓ. $\square$

Now, we are ready to calculate the minimum length of the suffix needed for σ to be consistent with a given concave* function f.

Lemma 2. *For any concave* function f, there exists an $n_0 \in \mathbb{N}$ such that the sequence $\sigma(n,m)$ is consistent with f, as long as $n \geq n_0$ and*

$$m \geq \frac{k+1-f(k+1)}{f(k+1)-1}\,(k+1)\,n + \frac{k^2}{f(k+1)-1} \quad .$$

Proof. Assume that $m \geq \frac{k+1-f(k+1)}{f(k+1)-1}\,(k+1)\,n + \frac{k^2}{f(k+1)-1}$ and let $A(\ell)$ be defined as in Lemma 1, and let $\ell' = f^{-1}(k+1)$. If $f(k+1) = k+1$, then $f(\ell) = \ell$, $1 \leq \ell \leq k+1$, since $f(1) = 1$ and $f(\ell+1)-f(\ell) \leq 1$ for all ℓ. In this case, $\text{Av}(\ell) \leq f(\ell)$, for all ℓ. Otherwise, $m \geq qn$, where $q > 0$ is independent on n, as required in Lemma 1. Thus, $\text{Av}(\ell) \leq A(\ell)$, for all ℓ. Moreover,

$A(1) = 1 = f(1)$ and $A(f^{-1}(k+1)) = k+1$. Thus, since f is concave, it suffices to prove that $A(k+1) \le f(k+1)$. This is done using algebraic manipulations:

$$\begin{aligned} A(k+1) \le f(k+1) \;&\Leftrightarrow\; k+1 - \frac{k(m-k)}{(k+1)n+m} \le f(k+1) \\ &\Leftrightarrow\; m \ge \frac{k+1-f(k+1)}{f(k+1)-1}\,(k+1)\,n + \frac{k^2}{f(k+1)-1} \enspace . \end{aligned}$$

□

Lemma 3. *Let $\sigma'(n,m)$ be any request sequence consisting of a prefix of $n(k+1)$ requests to pages from $\{p_1,\ldots,p_{k+1}\}$ and a suffix of m requests to the page p_1. Then for $1 \le \ell \le n(k+1)+m$, $\mathrm{Av}_{\sigma'}(\ell) \le \mathrm{Av}_{\sigma}(\ell)$.*

Proof. Both sequences have the same length, so it suffices to show that in all corresponding windows the sequence σ' can not have more distinct pages than σ. For $1 \le i \le (k+1)(n-1)+1$, σ has $N_\ell(i) = \min\{\ell, k+1\}$ which is the maximum possible number of distinct pages for window length ℓ. Hence, σ' cannot have more distinct pages in its corresponding window. For $(k+1)(n-1)+2 \le i \le (k+1)n$, observe that the $(k+1)$ requests $\sigma[(k+1)(n-1)+2],\ldots,\sigma[(k+1)n+1]$ are all distinct. Thus, $\sigma'_\ell(i)$ cannot have more distinct pages in a window starting in this range. For $i \ge (k+1)n+1$, $\sigma[i]$ and $\sigma'[i]$ are identical and there is nothing to prove for windows starting at $\sigma[i]$. □

Theorem 9. *For any deterministic online paging algorithm $\mathcal{A}$,*

$$F_{\mathcal{A}}(f) \ge \frac{f(k+1)-1}{k}.$$

Proof. Consider a request sequence of length $(k+1)n+m$ with $k+1$ distinct pages. Since the algorithm is deterministic and can hold only k distinct pages in its fast memory, we can choose the first $(k+1)n$ requests such that $\mathcal{A}$ incurs a page fault on every request. The remaining m requests all go to the page p_1. So $\mathcal{A}$ will have a least $(k+1)n$ page faults (assuming that the fast memory is initially empty). Let $m = \frac{k+1-f(k+1)}{f(k+1)-1}\,(k+1)\,n + \frac{k^2}{f(k+1)-1}$. Then, by Lemma 2 and Lemma 3, there exists an n_0 and a sequence of request sequences $\big(\sigma(n,m)\big)_{n \ge n_0}$ consistent with f and enforcing $(k+1)n$ page faults when serviced by $\mathcal{A}$. Thus, for $n \ge n_0$,

$$\begin{aligned} F_{\mathcal{A}}\big(\sigma(n,m)\big) &\ge \frac{n(k+1)}{|\sigma|} = \frac{n(k+1)}{n(k+1)+m} \\ &= 1/\left(1 + \frac{k+1-f(k+1)}{f(k+1)-1} + \frac{k^2}{n(k+1)(f(k+1)-1)}\right) \\ &= \frac{f(k+1)-1}{k + k^2/(n(k+1))} > \frac{f(k+1)-1}{k+k/n} \enspace . \end{aligned}$$

□

6.2 LRU and FIFO

It is easy to prove a tight upper bound for LRU. We shall use a technique that will be used in all of the proofs of upper bounds. Assume that the sequence contains n distinct pages $p_1,\ldots,p_n$. For each page p_i we shall derive a lower bound on the number $N_\ell(p_i)$ of windows of length ℓ containing p_i, $1 \le i \le n$, since $N_\ell = \sum_{i=1}^{n} N_\ell(p_i)$. First, we will show that LRU is optimal.

Theorem 10. *The fault rate of LRU is* $F_{LRU}(f) \leq \dfrac{f(k+1)-1}{k}$.

Proof. Consider a sequence σ consistent with f. When a page p is requested, none of the next k requests are faults on p. Thus, if $\sigma[i]$ is a request to the page p, the window $\sigma_{k+1}[i]$ does not contain a fault on p. Thus, for each page p, each fault on p is contained in $k+1$ windows of length $k+1$ containing no other faults on p and, for each request to p that does not produce a fault, there is at least one window of length $k+1$ that does not contain a fault on p and whose first request is a request to p. Thus, except for the first and last k requests, each fault producing request contributes $k+1$ to N_{k+1} and each request that does not produce a fault contributes at least one:

$$N_{k+1} \geq (k+1) \cdot \text{LRU}(\sigma) + \big(|\sigma| - \text{LRU}(\sigma)\big) - c = k \cdot \text{LRU}(\sigma) + |\sigma| - c,$$

where $c < 2k(k+1)$ is independent of $|\sigma|$. Thus,

$$\text{Av}(k+1) \geq \frac{k \cdot \text{LRU}(\sigma) + |\sigma| - c}{|\sigma|} = k \cdot F_{\text{LRU}}(\sigma) + 1 - \frac{c}{|\sigma|}.$$

Since σ is consistent with f,

$$f(k+1) \geq \text{Av}(k+1) \geq k \cdot F_{\text{LRU}}(\sigma) + 1 - \frac{c}{|\sigma|}.$$

Solving for $F_{\text{LRU}}(\sigma)$ yields the desired bound. □

To prove that FIFO is optimal in the Average-Model we need the following lemma.

Lemma 4. *For any request sequence σ and any ℓ, $1 \leq \ell \leq |\sigma|$, N_ℓ grows with at least 1, if a request is inserted in σ.*

Proof. Assume that the new request r is inserted in σ just after $\sigma[i-1]$ (see Figure 4), and let σ' denote the resulting request sequence. For $1 \leq j \leq i-\ell$, $\sigma_\ell[j] = \sigma'_\ell[j]$. For $i \leq j \leq |\sigma| - \ell + 1$, $\sigma_\ell[j] = \sigma'_\ell[j+1]$. Thus, it is sufficient to prove that

$$\sum_{j=i-\ell+1}^{j_{\max}} \Big(N'_\ell(j) - N_\ell(j)\Big) + N'_\ell(j_{\max}+1) \geq 1,$$

where $j_{\max} = \min\{i-1, |\sigma| - \ell + 1\}$.

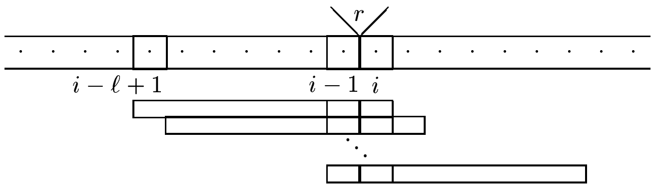


Figure 4: The windows $\sigma_\ell[i-\ell+1], \ldots, \sigma_\ell[i-1]$

Let $i-\ell+1 \leq j \leq j_{\max}$. Then $\sigma'_\ell[j]$ contains the request r and the requests in $\sigma_{\ell-1}[j]$. Therefore, $N'_\ell(j)$ and $N_\ell(j)$ can differ by at most 1. If $N'_\ell(j) < N_\ell(j)$, then the last page $\sigma[j+\ell-1]$ in $\sigma_\ell[j]$ is distinct from all pages in $\sigma_{\ell-1}[j]$, and r is contained in $\sigma_{\ell-1}[j]$. Thus, $\sigma[j+\ell-1] = \sigma'[j+\ell] \neq \sigma'[j+h]$, $0 \leq h \leq \ell - 1$.

Now, let S be the set consisting of the index i and each of the indices $j+\ell$ such that $N'_\ell(j) < N_\ell(j)$. For each pair $a, b \in S$, $\sigma'[a] \neq \sigma'[b]$. Thus,

$$N'_\ell(j_{\max}+1) \geq |S| = 1 + \sum_{j=i-\ell+1}^{j_{\max}} \left(N_\ell(j) - N'_\ell(j)\right).$$

□

Theorem 11. *The fault rate of FIFO is* $F_{FIFO}(f) \leq \dfrac{f(k+1)-1}{k}$.

Proof. Let σ be an arbitrary request sequence consistent with f. Let σ' be the subsequence of σ consisting only of the requests on which $\mathcal{A}$ produces a fault. Between two faults on a page p there are faults on at least k other pages. Thus, no window of length $k+1$ in σ' contains the same page twice. Therefore,

$$N'_{k+1} = (k+1)(|\sigma'| - k) = (k+1) \cdot \text{FIFO}(\sigma) - k(k+1).$$

By Lemma 4,

$$N_{k+1} \geq N'_{k+1} + \left(|\sigma| - \text{FIFO}(\sigma)\right) = k \cdot \text{FIFO}(\sigma) + |\sigma| - k(k+1).$$

Now, by the same arguments as in the proof of Theorem 10, the desired bound is obtained. □

6.3 Marking Algorithms

In this section we prove an upper bound on the fault rate of any marking algorithm of approximately $\frac{4}{3}\frac{f(k)}{k}$. Furthermore, we prove that there exists a marking algorithm and a concave* function for which the bound is tight.

Theorem 12. *For any Marking algorithm* $\mathcal{M}$,

$$F_{\mathcal{M}}(f) \leq \begin{cases} \dfrac{4k}{3k+2} \cdot \dfrac{f(k)}{k}, & \textit{if } k \textit{ is even,} \\ \dfrac{4k}{3k+2-1/k} \cdot \dfrac{f(k)}{k}, & \textit{if } k \textit{ is odd.} \end{cases}$$

Proof. Consider an arbitrary request sequence σ consistent with f. As a beginning, we will prove that $F_{\mathcal{M}}(\sigma) \leq \frac{4}{3}\frac{f(k)}{k}$. Analogous to the proof of Theorem 10, we will do this by proving $N_k \geq \frac{3k}{4}(\mathcal{M}(\sigma) - c)$, for some constant c (i.e., c is independent of the sequence length).

Partition σ into phases P^1, P^2, ... , P^n, such that each phase contains exactly k distinct pages (except for possibly the last phase) and the k pages are all different from the first page requested in the next phase. The pages of phase i, $1 \leq i \leq n-1$, $p_1^i, p_2^i, \ldots, p_k^i$ are numbered according to first appearance, i.e., the first page requested in P^i is p_1^i, the first page different from p_1^i is p_2^i, and so on. Each page produces at most one fault in the phase. For each phase P^i, let s_i denote the index of the first request in P^i, i.e., $\sigma[s_i] = p_1^i$.

For $2 \leq i \leq n-2$, let N_k^i denote $\sum_{j=s_i-\lceil\frac{k}{2}\rceil+1}^{s_{i+1}-\lceil\frac{k}{2}\rceil} N_k(j)$, and note that $N_k \geq \sum_{i=2}^{n-2} N_k^i$. Note that the first window contributing to N_k^i contains exactly $\lceil\frac{k}{2}\rceil - 1$ requests from phase P_{i-1} and the last window contains exactly $\lfloor\frac{k}{2}\rfloor$ requests from phase P_{i+1}. If the k distinct pages requested in P^i, $2 \leq i \leq n-2$, contribute at least $\frac{3k^2}{4}$ to N_k^i, then $N_k \geq \frac{3k^2}{4}(n-3) = \frac{3k}{4}(kn-3k) \geq \frac{3k}{4}(\mathcal{M}(\sigma)-3k)$.

Assume first that k is even. For $1 \le j \le \frac{k}{2}$, the first request to p_j^i is preceded by at least $j-1$ requests and succeeded by at least $\frac{k}{2}$ requests in the phase. Therefore, p_j^i is contained in at least $\frac{k}{2} - 1 + j$ windows contributing to N_k^i (see Figure 5).

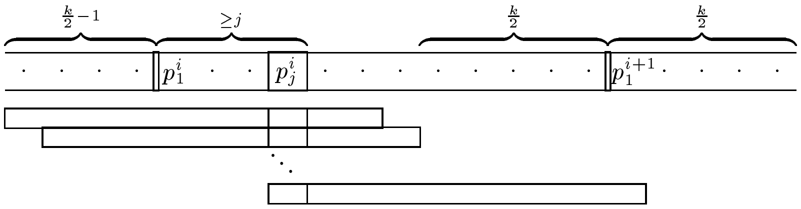


Figure 5: k even, $j \le \frac{k}{2}$: p_j^i is contained in at least $\frac{k}{2} - 1 + j$ windows contributing to N_k^i.

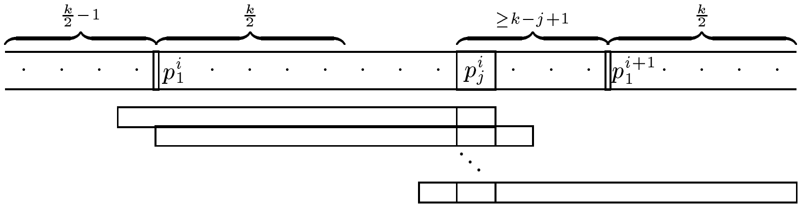


Figure 6: k even, $j \ge \frac{k}{2} + 1$: p_j^i is contained in at least $k - j + 1 + \frac{k}{2}$ windows contributing to N_k^i.

Similarly, for $\frac{k}{2} + 1 \le j \le k$, the first request to p_j^i is succeeded by at least $k - j$ requests and preceded by at least $\frac{k}{2}$ in the phase. Therefore, p_j^i is contained in at least $k - j + 1 + \frac{k}{2}$ windows contributing to N_k^i (see Figure 6). Thus,

$$\begin{aligned} N_k^i &\ge \sum_{j=1}^{\frac{k}{2}} \left(\frac{k}{2} - 1 + j\right) + \sum_{j=\frac{k}{2}+1}^{k} \left(\frac{3k}{2} - j + 1\right) \\ &= \sum_{j=0}^{\frac{k}{2}-1} \left(\frac{k}{2} + j\right) + \sum_{j=1}^{\frac{k}{2}} \left(\frac{k}{2} + j\right) = \frac{3k^2}{4} . \end{aligned}$$

This proves that $F_{\mathcal{M}}(\sigma) \le \frac{4}{3}\frac{f(k)}{k}$. To prove that $F_{\mathcal{M}}(\sigma) \le \frac{4k}{3k+2}\frac{f(k)}{k}$, it suffices to show that $N_k^i \ge \frac{3k+2}{4k} k^2 = \frac{3k^2}{4} + \frac{k}{2}$, $2 \le i \le n-2$. To do that, note that the first page p_1^{i+1} requested in phase P^{i+1} is not requested in P^i. Thus, p_1^{i+1} contributes $\frac{k}{2}$ to N_k^i.

Assume now that k is odd. For $1 \le j \le \frac{k-1}{2}$, p_j^i is contained in at least $\frac{k-1}{2} + j$ windows contributing to N_k^i. For $\frac{k+1}{2} \le j \le k$, p_j^i is contained in at least $k - j + 1 + \frac{k-1}{2}$ windows contributing to N_k^i. Thus,

$$N_k^i \ge \sum_{j=1}^{\frac{k-1}{2}} \left(\frac{k-1}{2} + j\right) + \sum_{j=1}^{\frac{k+1}{2}} \left(\frac{k-1}{2} + j\right) = \frac{3k^2}{4} + \frac{1}{4} .$$

To prove that $F_{\mathcal{M}}(\sigma) \le \frac{4k}{3k+2-\frac{1}{k}}\frac{f(k)}{k}$, it suffices to show that $N_k^i \ge \frac{3k+2-\frac{1}{k}}{4k} k^2 = \frac{3k^2}{4} + \frac{k}{2} - \frac{1}{4}$. This inequality holds, since p_1^{i+1} is contained in $\frac{k-1}{2}$ windows contributing to N_k^i. $\square$

For the lower bound, we make use of a sequence consisting of h distinct pages. Let $\text{UPDOWN}_h := \langle p_1, p_2, \ldots, p_{h-1}, p_h, p_{h-1}, \ldots, p_3, p_2 \rangle$ and let $\sigma = \text{UPDOWN}_h^n$ be the concatenation of n copies of UPDOWN_h. We refer to UPDOWN_h as a phase of σ and subdivide the phases into "up" and "down" sub-phases, each of length $h - 1$. Define $\text{Av}_h^\infty(\ell)$ to be the average number of distinct pages in windows of length ℓ in an infinitely long sequence UPDOWN_h^n, i.e., for $n \to \infty$. To calculate $\text{Av}_h^\infty(\ell)$ and prove that it is concave*, we shall need the following Lemma.

Lemma 5. *For* $1 \le \ell \le 2h - 3$, $\text{Av}_h^\infty(\ell + 1) - \text{Av}_h^\infty(\ell) = 1 - \dfrac{1}{h-1} \left\lfloor \dfrac{\ell}{2} \right\rfloor$.

Proof. Since the sequence has unbounded length, the average is the same in all its UPDOWN_h phases. Furthermore, averaging over a single "up" or a single "down" sub-phase gives the same result due to the symmetry of the sequence. We choose to analyze an "up" sub-phase.

Let N_ℓ^∞, $0 \le \ell \le 2h - 3$, be the sum of the number of distinct requests in all $h - 1$ windows of length ℓ starting within the considered "up" sub-phase. In order to prove the lemma, we show

$$(h-1)\big(\text{Av}_h^\infty(\ell+1) - \text{Av}_h^\infty(\ell)\big) = N_{\ell+1}^\infty - N_\ell^\infty = h - \lfloor \ell/2 \rfloor - 1.$$

Case $0 \le \ell \le h - 1$. Obviously, the first $h - \ell$ windows of length ℓ get a new page when lengthened by 1 position. Also some windows starting towards the end of the "up" sub-phase contribute a 1 to $N_{\ell+1}^\infty - N_\ell^\infty$. Precisely, for n odd, the last $\lfloor \ell/2 \rfloor$ windows get a new page and, for n even, there are $\ell/2 - 1$ windows of this kind. Thus,

$$N_{\ell+1}^\infty - N_\ell^\infty = h - \ell + \left\{ \begin{array}{l} \left\lfloor \dfrac{\ell}{2} \right\rfloor, \quad \ell \text{ odd} \\ \dfrac{\ell}{2} - 1, \ell \text{ even} \end{array} \right\} = \left\{ \begin{array}{l} h - \left\lceil \dfrac{\ell}{2} \right\rceil, \quad \ell \text{ odd} \\ h - \dfrac{\ell}{2} - 1, \ell \text{ even} \end{array} \right\} = h - \left\lfloor \dfrac{\ell}{2} \right\rfloor - 1.$$

Case $h \le \ell \le 2h - 3$. Again, we determine the number of windows that contribute 1 to the difference $N_{\ell+1}^\infty - N_\ell^\infty$. The first window cannot contribute a 1 because it already covers h distinct pages. Subsequent windows can only contribute if they are long enough to reach a new page in the following "down" sub-phase. Generally, the part of the window in the "down" sub-phase must be longer than the part in the "up" sub-phase. So only those windows starting at positions i, where

$$2(h-i) + 1 \le \ell \quad \Leftrightarrow \quad i \ge h - \frac{\ell - 1}{2}$$

can possibly contribute 1 to the difference. On the other hand, a window that starts in the "up" sub-phase and extends further than position $2h - 2$ (the end of the "down" sub-phase) cannot contribute a 1. So it must also hold that

$$i + (\ell - 1) \le 2h - 2 \quad \Leftrightarrow \quad i \le 2h - \ell - 1.$$

If ℓ is odd, there are $(2h - \ell - 1) - (h - \frac{\ell-1}{2}) + 1 = h - \frac{\ell+1}{2} = h - \lceil \frac{\ell}{2} \rceil = h - \lfloor \frac{\ell}{2} \rfloor - 1$ windows contributing a 1. Note that, for ℓ even, i must be at least $h - \frac{\ell-2}{2}$, since $i \in \mathbb{N}$. Thus, there are $(2h - \ell - 1) - (h - \frac{\ell-2}{2}) + 1 = h - \frac{\ell}{2} - 1 = h - \lfloor \frac{\ell}{2} \rfloor - 1$ contributing windows. □

Now we are ready to calculate $\text{Av}_h^\infty(\ell)$.

Lemma 6.

$$\mathrm{Av}_h^\infty(\ell) = \begin{cases} \ell - \dfrac{(\ell-1)^2}{4(h-1)}, & 1 \le \ell \le 2h-3,\ \ell \text{ odd,} \\ \ell - \dfrac{(\ell-1)^2-1}{4(h-1)}, & 2 \le \ell \le 2h-3,\ \ell \text{ even,} \\ h, & \ell \ge 2h-2, \end{cases}$$

and $\mathrm{Av}_h^\infty(\ell)$ *is concave**.

Proof. The equality follows from Lemma 5 and simple calculations. For $1 \le \ell \le 2h-3$,

$$\begin{aligned}\mathrm{Av}_h^\infty(\ell) &= \mathrm{Av}_h^\infty(1) + \sum_{i=1}^{\ell-1} \mathrm{Av}_h^\infty(i+1) - \mathrm{Av}_h^\infty(i) = 1 + \sum_{i=1}^{\ell-1}\left(1 - \frac{1}{h-1}\left\lfloor \frac{i}{2}\right\rfloor\right) \\ &= \ell - \frac{1}{h-1}\begin{cases} \dfrac{\ell-1}{2} + 2\displaystyle\sum_{i=1}^{(\ell-3)/2} i = \dfrac{(\ell-1)^2}{4}, & \ell \text{ odd,} \\ 2\displaystyle\sum_{i=1}^{\ell/2-1} i = \dfrac{(\ell-1)^2-1}{4}, & \ell \text{ even.}\end{cases}\end{aligned}$$

For $\ell \ge 2h-2$, each window of length ℓ contains all h pages and therefore, $\mathrm{Av}_h^\infty(\ell) = h$.

For the concave* property, it is obvious that $\mathrm{Av}_h^\infty(1) = 1$ and $\mathrm{Av}_h^\infty(\ell+1) - \mathrm{Av}_h^\infty(\ell) = 0$, for $\ell \ge 2h-2$. It remains only to check that

$$\forall \ell \in \{2,\dots,2h-2\}\colon 0 \le \mathrm{Av}_h^\infty(\ell+1) - \mathrm{Av}_h^\infty(\ell) \le \mathrm{Av}_h^\infty(\ell) - \mathrm{Av}_h^\infty(\ell-1) \le 1.$$

This is easily done using Lemma 5. For $2 \le \ell \le 2h-3$,

$$\begin{aligned}\mathrm{Av}_h^\infty(\ell+1) - \mathrm{Av}_h^\infty(\ell) &= 1 - \frac{1}{h-1}\left\lfloor\frac{\ell}{2}\right\rfloor \le 1 - \frac{1}{h-1}\left\lfloor\frac{\ell-1}{2}\right\rfloor \\ &= \mathrm{Av}_h^\infty(\ell) - \mathrm{Av}_h^\infty(\ell-1).\end{aligned}$$

Moreover,

$$\mathrm{Av}_h^\infty(2) - \mathrm{Av}_h^\infty(1) = 1$$

and

$$\begin{aligned}\mathrm{Av}_h^\infty(2h-2) - \mathrm{Av}(2h-3) &= 1 - \frac{1}{h-1}\left\lfloor\frac{2h-3}{2}\right\rfloor = 1 - \frac{h-2}{h-1} = \frac{1}{h-1} \\ &\ge \mathrm{Av}_h^\infty(2h-1) - \mathrm{Av}(2h-2) = 0.\end{aligned}$$

□

Lemma 7. *Let*

$$f(\ell) := \begin{cases} \min\{\ell, \mathrm{Av}_h^\infty(\ell) + \varepsilon\}, & 1 \le \ell \le 2h-3, \\ h, & \ell \ge 2h-2, \end{cases}$$

where $\varepsilon = \frac{h}{n-1}$. *Then,* $\textsc{UpDown}_h^n$ *is consistent with* f*, and* f *is concave**.

Proof. To prove that $\textsc{UpDown}_h^n$ is consistent with f, we must show that $\text{Av}(\ell) \le f(\ell)$, $1 \le \ell \le 2(h-1)n$. Obviously, for $\ell \ge 2h-2$, $f(\ell) = h$ is a tight upper bound on $\text{Av}(\ell)$. For $1 \le \ell \le 2h-3$, we utilize the results of Lemma 6. For the windows starting in one of the first $n-1$ phases of $\textsc{UpDown}_h^n$, the average number of distinct pages in a window of length ℓ is $\text{Av}_h^\infty(\ell)$. The sum of the number of distinct pages in all windows of length ℓ contained in the last $\textsc{UpDown}_h$ phase is at most $2(h-1)h$. Thus,

$$\begin{aligned}\text{Av}(\ell) &\le \frac{2(h-1)(n-1)\text{Av}_h^\infty(\ell) + 2(h-1)h}{n \cdot 2(h-1) - \ell + 1} \\ &\le \frac{2(h-1)(n-1)\text{Av}_h^\infty(\ell) + 2(h-1)h}{2(h-1)(n-1)} = \text{Av}_h^\infty(\ell) + \frac{h}{n-1}.\end{aligned}$$

It follows easily from Lemma 6 that f is concave*. □

Theorem 13. *There exists a Marking algorithm $\mathcal{M}^*$ and a concave* function f such that*

$$F_{\mathcal{M}^*}(f) \ge \begin{cases} \dfrac{4k}{3k+2} \cdot \dfrac{f(k)}{k}, & \text{if } k \text{ is even,} \\ \dfrac{4k}{3k+2-1/k} \cdot \dfrac{f(k)}{k}, & \text{if } k \text{ is odd.} \end{cases}$$

Proof of Theorem 13. Consider the sequence $\sigma = \textsc{UpDown}_{k+1}^n$, where $n > 0$ is a (large) integer, and the marking algorithm $\mathcal{M}^*$ that uses the LIFO (Last In First Out) strategy when evicting an unmarked page. Note that $\mathcal{M}^*$ will fault on every request in the sequence. Thus, $F_{\mathcal{M}^*}(\sigma) = 1$.

Let f be defined as in Lemma 7 with $h = k+1$. By Lemma 7, σ is consistent with f, and f is concave*. For $k \ge 3$, clearly, there exists an $n_0 \in \mathbb{N}$ such that $f(k) = \text{Av}_{k+1}^\infty(k) + \frac{k+1}{n-1}$, for $n \ge n_0$. Thus, for $k \ge 3$ and $n \ge n_0$, we can write the page fault rate in the following way.

$$\begin{aligned}F_{\mathcal{M}^*}(\sigma) = 1 &= \frac{k}{f(k)} \cdot \frac{f(k)}{k} = \frac{k}{\text{Av}_{k+1}^\infty(k) + \frac{k+1}{n-1}} \cdot \frac{f(k)}{k} \\ &= \begin{cases} \dfrac{4k}{3k+2-1/k+4\frac{k+1}{n-1}} \cdot \dfrac{f(k)}{k}, & k \text{ odd,} \\ \dfrac{4k}{3k+2+4\frac{k+1}{n-1}} \cdot \dfrac{f(k)}{k}, & k \text{ even.} \end{cases}\end{aligned}$$

□

6.4 The Optimal Offline Algorithm

In this section we will give an upper bound on the fault rate of LFD of approximately $\frac{4(M-k)}{4M-k}\frac{f(k+1)}{k+1}$. Recall that for any concave* function f, M denotes the maximum value of f. We will also prove that there exists a concave* function for which the bound is tight.

For the analysis of the upper bound, we will partition the sequences into phases $P^1, P^2, \ldots, P^n$ defined in the following way. The phase P^1 starts with the first request in the sequence. For $2 \le i \le n$, phase P^i starts with the first fault on a page that was evicted in phase P^{i-1}. Let s_i denote the index of the first request in P^i.

Unlike online algorithms, LFD never faults on every request of a long sequence. A request that is not a fault is called a *free* request. The proof of the upper bound given in Theorem 14 uses the following lemma.

Lemma 8. *For any free request r to some page p, let $W(r)$ be the number of windows of length $k+1$ containing r but no fault on p and no free request to p that occurs to the left of r. In each phase P^i, $2 \leq i \leq n-2$, there are at least $k-1$ free requests $r_1, r_2, \ldots, r_{k-1}$ to $k-1$ distinct pages such that*

$$\sum_{j=1}^{k-1} W(r_j) \geq W, \text{ where } W = \begin{cases} \frac{3}{4}k^2 - \frac{3}{4}, & k \text{ odd,} \\ \frac{3}{4}k^2 - 1, & k \text{ even.} \end{cases}$$

Proof. Let p be the first page requested in phase P^{i+1}. Then p is evicted at some point during phase P^i. Assume that this happens as a result of the request $\sigma[q]$. Since p is evicted, each of the $k-1$ other pages $p_1, \ldots, p_{k-1}$ in fast memory are requested at some point between $\sigma[q]$ and $\sigma[s_{i+1}]$. Each of these requests must be free, because otherwise P^i would contain a fault on a page that had been evicted earlier in the phase and this would contradict the definition of a phase. Thus, there are no faults on any of these $k-1$ pages after $\sigma[q]$ in phase P^i. For $1 \leq j \leq k-1$, let r_j be the first request to p_j after $\sigma[q]$.

For $1 \leq j \leq k-1$, let h_j be the index of r_j, i.e., $r_j = \sigma[h_j]$. It is clear that, for $1 \leq t \leq k$, $\sigma[h_j + t]$ is not a fault on p_j. Thus, to calculate $W(r_j)$ we only need to consider requests to p_j to the left of r_j. Let $d_{\mathrm{h}}(j) = h_j - q$. Similarly, let f_j be the largest index smaller than q such that $\sigma[f_j]$ is a request to p_j, and let $d_{\mathrm{f}}(j) = q - f_j$. Finally, let $D(j) = d_{\mathrm{f}}(j) + d_{\mathrm{h}}(j)$. Then $W(r_j) = \min\{k+1, D(j)\}$, since there are at least k pages in P^{i-1} and in P^{i+1}. Note that

$$\sum_{j=1}^{k-1} D(j) = \sum_{j=1}^{k-1} d_{\mathrm{f}}(j) + \sum_{j=1}^{k-1} d_{\mathrm{h}}(j) \geq 2\sum_{j=1}^{k-1} j.$$

Let S be the set of requests r_j such that $D(j) \leq k+1$, and let $m = |S|$. Then,

$$\begin{aligned} \sum_{j=1}^{k-1} W(r_j) &\geq (k-1-m)(k+1) + \sum_{r_j \in S} D(j) \\ &\geq k^2 - 1 - m(k+1) + 2\sum_{j=1}^{m} j = k^2 - 1 + m^2 - km. \end{aligned}$$

This lower bound on $\sum_{j=1}^{k-1} W(r_j)$ is minimized when $m = \frac{k}{2}$, if k is even, and when $m = \frac{k-1}{2}$, if k is odd. Inserting these values of m in the lower bound, the inequality of the lemma is obtained. □

Theorem 14. *The fault rate of LFD is*

$$F_{LFD}(f) \leq \begin{cases} \dfrac{4(M-k)}{4M-k-3} \cdot \dfrac{f(k+1)}{k+1}, & k \text{ odd} \\ \dfrac{4(M-k)}{4M-k-3-\frac{1}{k+1}} \cdot \dfrac{f(k+1)}{k+1}, & k \text{ even.} \end{cases}$$

Proof. Consider any request sequence σ consistent with f. Like in the proof of Theorem 10 we shall calculate how much each request contributes to N_{k+1}. Like in the case of LRU and FIFO, no window of length $k+1$ contains more than one fault on the same page. Thus, each fault contributes $k+1$. Lemma 8 provides a lower bound on the contribution from free requests. For each free request r to some page p we count the windows containing r. To ensure that nothing is counted twice, we consider only those windows that do not contain a fault on p. If a window contains two free requests to p contained in two distinct phases, then the window is only counted in the first of the two phases.

Within a phase there is at most one fault on each page, and the k pages that are in fast memory at the beginnning of a phase do not produce a fault within the phase. Thus, each phase contains at most $M-k$ faults. Let F^i be the number of faults in phase P^i, let W be defined as in Lemma 8, and let N^i_{k+1} be the contribution to N_{k+1} from the requests in P^i. Then

$$\frac{N^i_{k+1}}{F^i} \geq \frac{(k+1)F^i + W}{F^i} \geq \frac{(k+1)(M-k)+W}{M-k} \quad .$$

Solving for F^i yields

$$F^i \leq \frac{M-k}{(k+1)(M-k)+W} \cdot N^i_{k+1}, \text{ and}$$

$$\begin{aligned}
\text{LFD}(\sigma) \;&=\; \sum_{i=1}^{n} F^i \;=\; \sum_{i=2}^{n-2} F^i + c \;\leq\; \frac{M-k}{(k+1)(M-k)+W} \sum_{i=2}^{n-2} N^i_{k+1} + c \\
&=\; \frac{M-k}{(k+1)(M-k)+W} \cdot N_{k+1} + c',
\end{aligned}$$

where c and c' are constants, i.e., independent on $|\sigma|$. Thus,

$$\begin{aligned}
F_{\text{LFD}}(\sigma) &\leq \frac{M-k}{(k+1)(M-k)+W} \cdot \text{Av}(k+1) + \frac{c'}{|\sigma|} \\
&\leq \frac{M-k}{(k+1)(M-k)+W} \cdot f(k+1) + \frac{c'}{|\sigma|} \quad .
\end{aligned}$$

Now, the theorem follows by using that $\frac{3}{4}k^2 - \frac{3}{4} = \frac{3}{4}(k-1)(k+1)$.

$$\begin{aligned}
F_{\text{LFD}}(\sigma) &\leq \frac{M-k}{(k+1)(M-k)+\frac{3}{4}(k-1)(k+1)}\, f(k+1) + \frac{c'}{|\sigma|} \\
&= \frac{4(M-k)}{4M-k-3}\, \frac{f(k+1)}{k+1} + \frac{c'}{|\sigma|}, \; k \text{ odd, and}
\end{aligned}$$

$$\begin{aligned}
F_{\text{LFD}}(\sigma) &\leq \frac{M-k}{(k+1)(M-k)+\frac{3}{4}(k-1)(k+1)-\frac{1}{4}}\, f(k+1) + \frac{c'}{|\sigma|} \\
&= \frac{4(M-k)}{4M-k-3-\frac{1}{k+1}}\, \frac{f(k+1)}{k+1} + \frac{c'}{|\sigma|}, \; k \text{ even.}
\end{aligned}$$

□

Theorem 15. *There exists a concave* function f such that*

$$F_{LFD}(f) \geq \begin{cases} \dfrac{4(M-k)}{4M-k-3} \cdot \dfrac{f(k+1)}{k+1}, & k \text{ odd} \\[2ex] \dfrac{4(M-k)}{4M-k-3-\frac{1}{k+1}} \cdot \dfrac{f(k+1)}{k+1}, & k \text{ even.} \end{cases}$$

Proof. Consider the function f given in Lemma 7. For $k \geq 3$ and n sufficiently large, inserting $h = M$ yields,

$$\begin{aligned}
f(k+1) &= k+1-\frac{k^2-1}{4(M-1)}+\frac{M}{n-1} \\
&= \frac{4(M-1)(k+1)}{4(M-1)}-\frac{(k-1)(k+1)}{4(M-1)}+\frac{M}{n-1} \\
&= \frac{(4M-k-3+\varepsilon)(k+1)}{4(M-1)}, \qquad k \text{ odd, and} \\
f(k+1) &= k+1-\frac{k^2}{4(M-1)}+\frac{M}{n-1} \\
&= \frac{4(M-1)(k+1)}{4(M-1)}-\frac{(k-1)(k+1)+1}{4(M-1)}+\frac{M}{n-1} \\
&= \frac{(4M-k-3-\frac{1}{k+1}+\varepsilon)(k+1)}{4(M-1)}, \qquad k \text{ even,}
\end{aligned}$$

where $\varepsilon = \frac{4(M-1)}{k+1}\frac{M}{n-1}$. The sequence $\textsc{UpDown}_M^n$ is consistent with f and f is concave*. It is easy to verify that, in each "up" and each "down" sub-phase, LFD faults on the first request and the last $M-k-1$ requests . Thus,

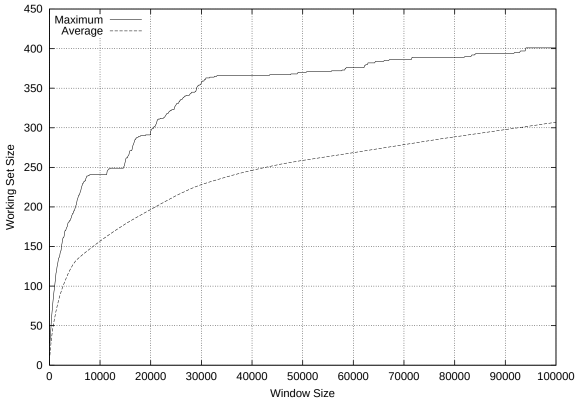
$$\begin{aligned}
F_{\text{LFD}}(\textsc{UpDown}_M^n) &= \frac{M-k}{M-1}\cdot\frac{f(k+1)}{f(k+1)} \\
&= \begin{cases} \frac{M-k}{M-1}\cdot\frac{4(M-1)}{4M-k-3+\varepsilon}\cdot\frac{f(k+1)}{k+1}, & k \text{ odd} \\ \frac{M-k}{M-1}\cdot\frac{4(M-1)}{4M-k-3-\frac{1}{k+1}+\varepsilon}\cdot\frac{f(k+1)}{k+1}, & k \text{ even} \end{cases} \\
&= \begin{cases} \frac{4(M-k)}{4M-k-3+\varepsilon}\cdot\frac{f(k+1)}{k+1}, & k \text{ odd} \\ \frac{4(M-k)}{4M-k-3-\frac{1}{k+1}+\varepsilon}\cdot\frac{f(k+1)}{k+1}, & k \text{ even.} \end{cases}
\end{aligned}$$

□

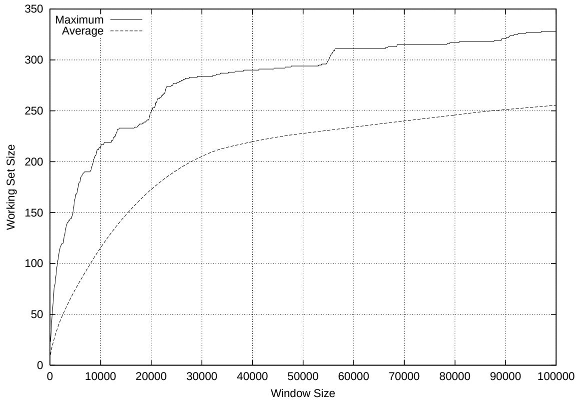
7 Experiments

In this section we present some results of our experimental study in which we compared the worst-case fault rates developed in the previous sections to the fault rates observed on real processor traces. We analyzed memory reference traces from the New Mexico State University Trace Base [14] that contains standard benchmarks. We selected traces from VAX and SPARC platforms. More specifically, we chose the ATUM VAX traces and a bundle of SPARC traces that were collected while running the SPEC92 benchmark suite. The sets consist of a collection of 9 respectively 13 memory reference traces from single processes. The request sequences contain both data read/write requests and instruction fetches. The SPARC traces were truncated after 10 million references, whereas the VAX traces vary in length, but are all about $400{,}000$ requests. We worked with a page size of 512 bytes for the VAX architecture and a page size of 2048 bytes for the SPARC architecture.

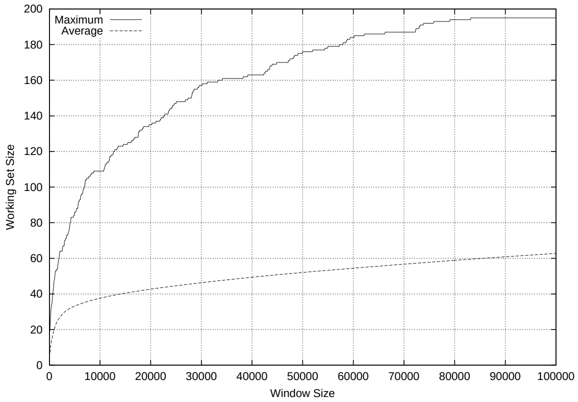
We first analyzed the maximum and average working set size in windows of up to $100{,}000$ requests. Figure 7 presents the results for four specific traces, two VAX traces and two SPARC traces.



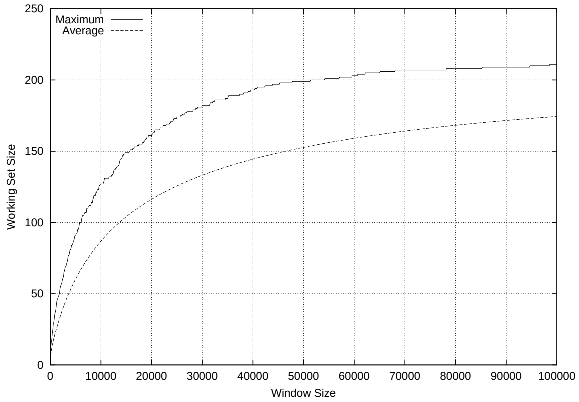
(a) VAX, PASCAL, 500 pages.



(b) VAX, SPIC, 385 pages.

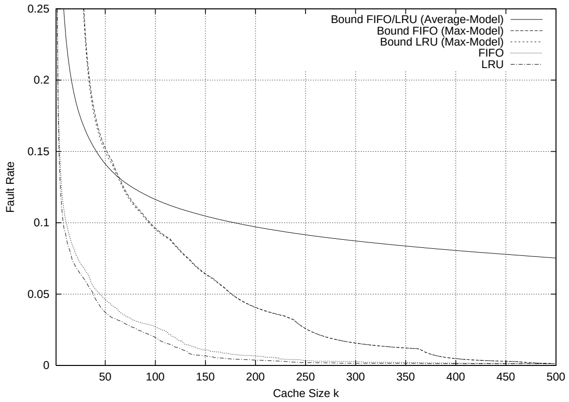


(c) SPARC, GCC, 276 pages.

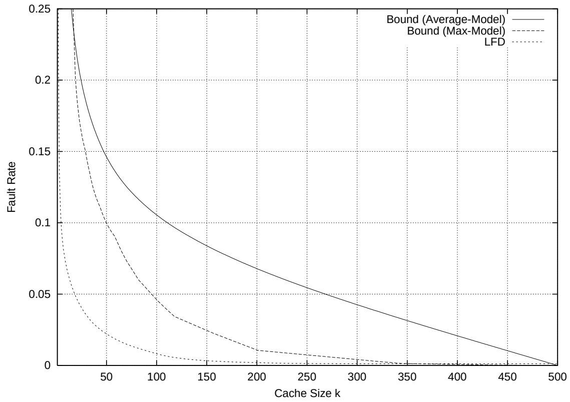


(d) SPARC, COMPRESS, 229 pages.

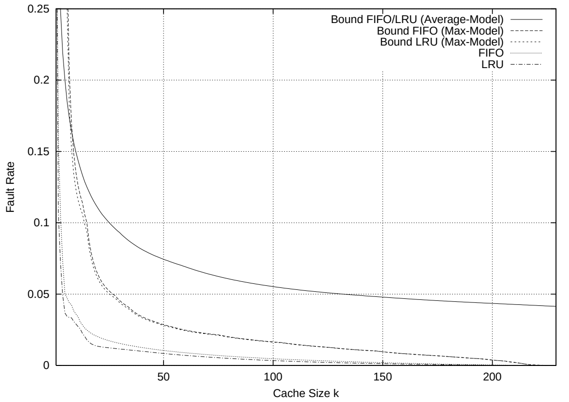
Figure 7: Maximum and average size of the working set in windows of size up to 100,000 requests. Each diagram's caption gives the architecture, the name of the trace, and the number of distinct pages requested in the entire sequence.



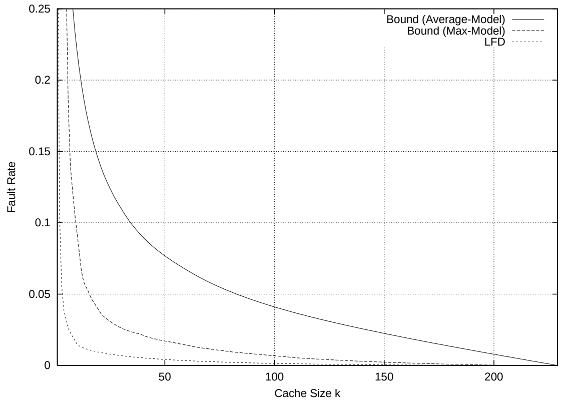
(a) VAX, PASCAL, FIFO and LRU.



(b) VAX, PASCAL, LFD



(c) SPARC, COMPRESS, FIFO and LRU.



(d) SPARC, COMPRESS, LFD.

Figure 8: Measured fault rates and upper bounds on the fault rates for FIFO, LRU and LFD. The fast memory size k varies in the range of 1 up to the total number of distinct pages requested in the entire sequence.

As illustrated by the figure, the behavior of the working set size proposed by Denning for a single window of increasing size can also be observed *globally*, taking the maximum/average working set size over *all* windows of a request sequence; the curves have an overall concave behavior. Only in the Max-Model, some minor adjustments are necessary to obtain a concave* function. We also observe that, for all window sizes, the working set size is very small.

In the second part of the experiments, we evaluated the fault rates of LRU, FIFO, and LFD on the various traces and compared the values to the corresponding bounds we developed for both the Max- and the Average-Model. We performed the comparison for cache sizes ranging from 1 to the maximum working set size. Figure 8 presents the results for the VAX Pascal and the SPARC Compress traces. The left part of the figure shows the results for LRU and FIFO. The two lower curves represent the empirical fault rates of LRU and FIFO, while the two curves in the middle show the corresponding theoretical upper bounds in the MAX-Model. The upper curve depicts the bound in the Average-Model. The right part of Figure 8 shows the bounds for LFD in the same relative order.

Since the fault rate as defined in Definition 2 is a worst-case measure, we cannot expect that the theoretical bounds on the fault rates match the empirical values completely. Nevertheless, the gap is not large and *considerably* smaller than in the case of competitiveness. On real world traces, the "empirical competitiveness" of LRU and FIFO is typically no larger then 4. This was observed in

[3, 18] and also showed in our experiments. On the other hand, the competitive ratios from theory are k. Thus, the gap between the theoretical and empirical competitiveness is $k/4$. In our paging model, the gaps are considerably smaller. For the SPARC COMPRESS trace the gap is, expressed as a function linear in k, usually between $k/50$ to $k/30$. For some of the traces we examined, the values were even below $k/1000$. We also remark that the results for the Max-Model are better than for the Average-Model. We conclude that while the Average-Model is interesting from a mathematical point of view, the Max-Model more accurately models request sequences that occur in practice.

References

[1] L.A. Belady. A study of replacement algorithms for virtual storage computers. *IBM Systems Journal*, 5:78-101, 1966.

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

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

[4] M. Chrobak and J. Noga. LRU is better than FIFO. *Proc. 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, 78–81, 1998.

[5] H.M. Deitel. *Operating Systems.* Addison-Wesley, 1990.

[6] P.J. Denning. The working set model of program behavior. *Communications of the ACM*, 11:323–333, 1968.

[7] P.J. Denning. Working sets past and present. *IEEE Transactions on Software Engineering*, 6:64–84, 1980.

[8] A. Fiat and A. Karlin. Randomized and multipointer paging with locality of reference. *Proc. 27th Annual ACM Symposium on Theory of Computing*, 626–634, 1995.

[9] A. Fiat and M. Mendel. Truly online paging with locality of reference. *Proc. 38rd Annual Symposium on Foundations of Computer Science*, 326–335, 1997.

[10] P.A. Franaszek and T.J. Wagner. Some distribution-free aspects of paging algorithm performance. *Journal of the ACM*, 21:31–39, 1974.

[11] S. Irani, A.R. Karlin and S. Phillips. Strongly competitive algorithms for paging with locality of reference. *SIAM Journal on Computing*, 25:477-497, 1996.

[12] A. Karlin, S. Phillips und P. Raghavan. Markov paging. *Proc. 33rd Annual Symposium on Foundations of Computer Science*, 208–216, 1992.

[13] E. Koutsoupias and C.H. Papadimitriou. Beyond competitive analysis. *Proc. 35th Annual Symposium on Foundations of Computer Science* 394–400, 1994.

[14] New Mexico State University. Homepage of New Mexico State University TraceBase (Online). Available: http://tracebase.nmsu.edu/tracebase.html.

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

[16] A.S. Tanenbaum. *Modern Operating System.* Prentice Hall, 1992.

[17] E. Torng. A unified analysis of paging and caching. *Algorithmica*, 20:175-200, 1998.

[18] N.E. Young. The k-server dual and loose competitiveness for paging. *Algorithmica*, 11:525–541, 1994.