# On Certain New Models for Paging with Locality of Reference

Reza Dorrigiv and Alejandro López-Ortiz

Cheriton School of Computer Science,
University of Waterloo,
Waterloo, ON, N2L 3G1, Canada,
{rdorrigiv, alopez-o}@uwaterloo.ca

**Abstract.** The competitive ratio is the most common metric in online algorithm analysis. Unfortunately, it produces pessimistic measures and often fails to distinguish between paging algorithms that have vastly differing performance in practice. An apparent reason for this is that the model does not take into account the locality of reference evidenced by actual input sequences. Therefore many alternative measures have been proposed to overcome the observed shortcomings of competitive analysis in the context of paging algorithms. While a definitive answer to all the concerns has yet to be found, clear progress has been made in identifying specific flaws and possible fixes for them. In this paper we consider two previously proposed models of locality of reference and observe that even if we restrict the input to sequences with high locality of reference in them the performance of every on-line algorithm in terms of the competitive ratio does not improve. Then we prove that locality of reference is useful under some other cost models, which suggests that a new model combining aspects of both proposed models can be preferable. We also propose a new model for locality of reference and prove that the randomized marking algorithm has better fault rate on sequences with high locality of reference. Finally we generalize the existing models to several variants of the caching problem.

## 1 Introduction

The competitive ratio is the most common metric in on-line algorithm analysis. Formally introduced by Sleator and Tarjan, it has served as a practical framework for the study of algorithms that must make irrevocable decisions in the presence of only partial information [11]. On-line algorithms are more often than not amenable to analysis under this framework; that is, computing the competitive ratio has proven to be effective—even in cases where the exact shape of the optimal solution is unknown. On the other hand, there are known applications in which the competitive ratio produces somewhat unsatisfactory results. In some cases it results in unrealistically pessimistic measures; in others, it fails to distinguish between algorithms that have vastly differing performance under any practical characterization.

**Paging.** The paging problem is an important on-line problem both in theory and in practice. In this problem we have a small but fast memory (cache) of size $k$ and a larger slow memory. The input is a sequence of page requests. The on-line paging algorithm should serve the requests one after another. For each request, if the requested page is in the cache, a hit occurs and the algorithm can serve the request without incurring any cost. Otherwise a fault occurs and the algorithm should bring the requested page to the cache. If the cache is already full, the algorithm should evict at least one page in order to make room for the new page. The objective is to design efficient on-line algorithms in the sense that on a given request sequence the total cost, namely the total number of faults, is kept low. Three well known paging algorithms are *Least-Recently-Used* (LRU), *First-In-First-Out* (FIFO), and *Flush-When-Full* (FWF). On a fault, if the cache is full, LRU evicts the page that is least recently requested, FIFO evicts the page that is first brought to the cache, and FWF empties the cache.

A paging algorithm is called *conservative* if it incurs at most $k$ page faults on any page sequence that contains at most $k$ distinct pages. LRU and FIFO are conservative while FWF is not [3]. Another important class of on-line paging algorithms is *marking* algorithms. A marking algorithm $\mathcal{A}$ works in phases. Each phase consists of the maximal sequence of requests that contain at most $k$ distinct pages. All the pages in the cache are unmarked at the beginning of each phase. We mark any page just after the first request to it. When an eviction is necessary, $\mathcal{A}$ should evict an unmarked page until none exists, which marks the end of the phase. LRU and FWF are marking algorithms while FIFO is not [3]. We only consider *demand paging* algorithms, i.e., algorithms that do not evict any pages on a hit. Any paging algorithm can be modified into a demand paging algorithm that has no more faults [3].

The cost of a paging algorithm $\mathcal{A}$ on an input sequence $I$ is the number of faults it incurs to serve $I$. In competitive analysis, we compare on-line algorithms to the off-line optimal algorithm $OPT$ which knows the entire sequence in advance. An on-line algorithm $\mathcal{A}$ is said to have competitive ratio $c$ if $\mathcal{A}(I) \leq c \times OPT(I)$ for all input sequences $I$.

**Alternative Measures.** It is known that the competitive ratio of any deterministic on-line paging algorithm is at least $k$ and the competitive ratio of any conservative or marking paging algorithm is at most $k$ (see e.g. [3]). This means that all conservative and marking algorithms are optimal with respect to the competitive ratio. Therefore competitive analysis cannot distinguish among the algorithms LRU, FIFO, and FWF. However, LRU is preferable in practice and furthermore its practical performance ratio is much better than its competitive ratio. These drawbacks of the competitive ratio have led to several proposals for better performance measures. For a survey on alternative measures see the monograph by Dorrigiv and López-Ortiz [6].

One reason that LRU has good experimental behaviour is that in practice page requests show *locality of reference*. This means that when a page is requested it is more likely to be requested in the near future. Most measures for the analysis of the paging algorithms try to use this fact by restricting their legal inputs to those with high locality. Several models have been suggested for paging with locality of reference (e.g. [4, 9, 12, 1, 10, 2]).

In this paper we consider the models proposed by Torng [12] and Albers et al. [1]. These models are based on the Denning's working set model [5] and we term them the *k-phase model* and the *working set model*, respectively. In the $k$-phase model, we consider decompositions of input sequences to phases in the same way as marking algorithms. For an input sequence $I$, let $D(I,k)$ be the decomposition of $I$ into phases of a marking algorithm with cache of size $k$, $|D(I,k)|$ be the number of phases of $D(I,k)$, and $L(I,k) = |I|/|D(I,k)|$ be the average length of phases. We call $I$ $a$-local if $L(I,k) \geq a \cdot k$. Torng models locality by restricting the input to $a$-local sequences for some constant $a$. In the working set model, a request sequence has high locality of reference if the number of distinct pages in a window of size $n$ is small. For a concave function $f$, we say that a request sequence is consistent with $f$ if the number of distinct pages in any window of size $n$ is at most $f(n)$, for any $n \in \mathcal{N}$. In this way, Albers et al. model locality by restricting the input to sequences that are consistent with a concave function $f$.

Torng shows that marking algorithms perform better on sequences with high locality under the *full access cost model*. In this model, the cost of a hit is one and the cost of a fault is $p+1$, where $p$ is a parameter of model that is called the *miss penalty*. We denote the cost of an algorithm $\mathcal{A}$ on a sequence $I$ in the full access cost model by $\mathcal{A}_{FA}(I)$. Albers et al. use the *fault rate* as their performance measure. The fault rate of an algorithm $\mathcal{A}$ on an input sequence $I$, denoted by $F_{\mathcal{A}}(I)$, is defined as $\mathcal{A}(I)/|I|$, where $|I|$ is the length of $I$.

**Our Results.** First we formally show that under the standard cost model the competitive ratio of every on-line paging algorithm remains the same under the Albers et al. and the Torng locality of reference models. Hence, new results about paging algorithms necessitate a change to the cost model. We apply the fault rate cost model to the $k$-phase model and the full access cost model to the working set model. These two had been previously studied under the alternate combination. The full access cost model compares on-line algorithms to the optimal off-line algorithm, while the fault rate cost model does not. Therefore there is not a direct relationship between the two cost models and our results cannot be directly concluded from the results of Torng [12] and Albers et al. [1]. Recently, Angelopoulos et al. [2] considered the standard cost model (number of faults) together with a new comparison model, called bijective analysis, and proved that LRU is the best on-line paging algorithm on sequences that show locality of reference in the working set model. Furthermore, we propose a new model for locality of reference and show that the randomized marking algorithm of [7] benefits from the locality of reference assumption. Finally we apply the locality of reference assumption to the caching problem, a generalization of the paging problem in which pages have different sizes and retrieval costs. We extend the existing models of locality of reference and show that certain caching algorithms perform better on sequences with good locality of reference under this model.

This paper is part of a systematic study of alternative models for paging. It contrasts two previously proposed models under locality of reference assumptions. Then using lessons learned it introduces a new model and shows that this new model is able to reflect locality of reference assumptions for the randomized marking algorithm. While the ultimate model for paging analysis remains yet to be discovered, the lessons learned (both positive and negative) from the results in this paper are of value to the field, and to judge from the past, likely to be of use in further future refinements in the quest for the ultimate model for paging analysis.

## 2 Limitations of the Competitive Ratio Model

We prove that restricting input sequences to those with high locality of reference is not reflected as an improvement on the competitive ratio.

**Observation 1** *If we restrict the input to sequences that are consistent with a concave function $f$, the competitive ratio of deterministic on-line paging algorithms does not improve.*

*Proof.* The proof idea is the same as the one used to show that finite lookahead does not improve competitive ratio of on-line paging algorithms [3]. Let $\mathcal{A}$ be a deterministic paging algorithm, we can obtain a worst case sequence by requesting always a page which is not in the cache. To ensure that requesting such a page is consistent with $f$, we repeat the last request of $I$ a sufficient number of times. Since we only consider demand paging algorithms, this has no effect on the contents of the cache. □

A similar result for the $k$-phase model can be proven, as we can make any sequence $a$-local by repeating each request $a$ times.

**Observation 2** *If we restrict the input to $a$-local sequences, the competitive ratio of deterministic on-line paging algorithms does not change.*

## 3 The Fault Rate of the $k$-phase Model

We obtain new results based on the $k$-phase model by considering the fault rate as the cost model. The fault rate of an algorithm $\mathcal{A}$ on $a$-local sequences is defined as $F_{\mathcal{A}}(a) = \inf\{r \mid \exists\, n \in \mathcal{N} : \forall I,\, L(I,k) \geq ak,\, |I| \geq n : F_{\mathcal{A}}(I) \leq r\}$. We also denote by $F_{\mathcal{A}}(I)$ the fault rate of $\mathcal{A}$ on an arbitrary sequence $I$. We obtain the following bound for the fault rate of marking algorithms.

**Theorem 1.** *Let $\mathcal{A}$ be an arbitrary marking algorithm and $a > 0$ be a constant. Then $F_{\mathcal{A}}(a) \leq 1/a$.*

*Proof.* Consider an arbitrary $a$-local sequence $I$ of size at least $n$. We show that $F_{\mathcal{A}}(I) \leq 1/a$. Consider the decomposition $D(I,k)$ of $I$. $\mathcal{A}$ does not fault more than once on a page $P$ in a phase $\phi$ because after the first fault, it marks $P$ and does not evict it in the reminder of $\phi$. Since each phase contains at most $k$ distinct pages, $\mathcal{A}$ does not fault more than $k$ times in a phase.

Thus $\mathcal{A}$ incurs at most $k \cdot |D(I,k)|$ faults on $I$ and we have $F_{\mathcal{A}}(I) \leq \frac{k \cdot |D(I,k)|}{|I|}$. Using $L(I,k) = |I|/|D(I,k)|$, we get $F_{\mathcal{A}}(I) \leq \frac{k}{L(I,k)} \leq \frac{k}{ak} = 1/a$. □

This theorem shows that the fault rate of any marking algorithm decreases as the locality of reference of the input increases. Note that this holds for every algorithm $\mathcal{A}$ that incurs at most $k$ faults in each phase. Since any phase contains at most $k$ distinct pages, we obtain the following result.

**Corollary 1.** *Let $\mathcal{A}$ be an arbitrary conservative algorithm and $a > 0$ be a constant. Then $F_{\mathcal{A}}(a) \leq 1/a$.*

## 4 The Working Set Model under Full Access Cost Model

In this section we apply the full access cost model to the working set model. Earlier we proved that the standard competitive ratio does not improve for sequences with high locality of reference in this model. Now we show that the competitive ratio of the classical algorithms in the full access cost model improves for such sequences.

First we use some results of Albers et al. [1] about the fault rate of paging algorithms. These results are expressed in term of $f^{-1}$, the inverse function of $f$, defined as

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

In other words, $f^{-1}(m)$ denotes the minimum size of a window that contains at least $m$ distinct pages.

**Theorem 2.** *The competitive ratio of LRU with respect to a concave function $f$ in the full access cost model is at most $\frac{p \cdot k \cdot (k-1) + k(f^{-1}(k+1)-2)}{p \cdot (k-1) + k(f^{-1}(k+1)-2)}$.*

*Proof.* Albers et al. proved that the fault rate of LRU is at most $\frac{k-1}{f^{-1}(k+1)-2}$ [1]. Consider an arbitrary sequence $I$ that is consistent with $f$. Suppose that LRU (OPT) incurs $m$ ($m'$) faults on $I$. We have $\frac{m}{|I|} \leq \frac{k-1}{f^{-1}(k+1)-2} \implies |I| \geq \frac{m \cdot (f^{-1}(k+1)-2)}{k-1}$. Since $m' \geq m/k$, we have $OPT_{FA}(I) = p \cdot m' + |I| \geq p \cdot m/k + |I|$. We also have $LRU_{FA}(I) = p \cdot m + |I|$, and therefore $\frac{LRU_{FA}(I)}{OPT_{FA}(I)} \leq \frac{p \cdot m + \frac{m \cdot (f^{-1}(k+1)-2)}{k-1}}{p \cdot m/k + \frac{m \cdot (f^{-1}(k+1)-2)}{k-1}} = \frac{p \cdot k \cdot (k-1) + k(f^{-1}(k+1)-2)}{p \cdot (k-1) + k(f^{-1}(k+1)-2)}$. Since $I$ was an arbitrary sequence, this proves the theorem. □

As the cost of a fault $p$ increases, the upper bound of Theorem 2 approaches $k$. When $p$ is not too large, the term $(f^{-1}(k+1)-2)$ becomes important. For a fixed $p$, the larger the value of $f^{-1}(k+1)$, the better the upper bound of the theorem. This supports our intuition that LRU has better performance on sequences with more locality of reference.

It is also known that $F_{FIFO}(f) \leq \frac{k}{f^{-1}(k+1)-1}$ and $F_{\mathcal{A}}(f) \leq \frac{k}{f^{-1}(k+1)-1}$ for any marking algorithm $\mathcal{A}$ [1]. We can use these results to prove the following theorem in an analogous way to Theorem 2.

**Theorem 3.** *Let $\mathcal{A}$ be a marking algorithm or FIFO. The competitive ratio of $\mathcal{A}$ with respect to a concave function $f$ in the full access cost model is at most* $\frac{p\cdot k+(f^{-1}(k+1)-1)}{p+(f^{-1}(k+1)-1)}$.

Finally we prove a result for all marking and conservative algorithms.

**Theorem 4.** *Let $\mathcal{A}$ be a marking or conservative algorithm. The competitive ratio of $\mathcal{A}$ with respect to a concave function $f$ in the full access cost model is at most* $\frac{p\cdot k+f^{-1}(k)}{p+f^{-1}(k)}$.

*Proof.* Let $I$ be a sequence consistent with $f$ and consider the decomposition $D(I,k)$ of $I$. We know that $\mathcal{A}$ incurs at most $k$ faults in each phase. Let $m$ denote the number of faults $\mathcal{A}$ incurs on $I$. We have $m \leq k \cdot |D(I,k)| \Rightarrow |D(I,k)| \geq m/k$. Each phase has length at least $f^{-1}(k)$ because $I$ is consistent with $f$. Therefore $|I| \geq |D(I,k)| \cdot f^{-1}(k) \geq m \cdot f^{-1}(k)/k$, and $\frac{\mathcal{A}_{FA}(I)}{OPT_{FA}(I)} \leq \frac{p\cdot m+|I|}{p\cdot m/k+|I|} \leq \frac{p\cdot m+m\cdot f^{-1}(k)/k}{p\cdot m/k+m\cdot f^{-1}(k)/k} = \frac{p\cdot k+f^{-1}(k)}{p+f^{-1}(k)}$. □

## 5 A New Model for Locality of Reference

In this section we introduce a new model for locality of reference that can be used to show that the *randomized marking* algorithm [7] benefits from locality of reference. We can generalize the definitions of the competitive ratio and the fault rate to the randomized algorithms by considering the expected number of page faults. Let $H_k$ denote the $k^{th}$ harmonic number: $H_k = 1 + 1/2 + \ldots + 1/k$. Fiat et al. introduced the randomized marking algorithm, $\mathcal{RM}$, and showed that it is $2H_k$-competitive [7]. The phases of $\mathcal{RM}$ are defined as deterministic marking algorithms. On a fault, $\mathcal{RM}$ evicts a page chosen uniformly at random from among the unmarked pages. A page is called *clean* if it was not requested in the previous phase and *stale* otherwise. Intuitively, a sequence with locality of reference does not have many clean pages in a phase. In order to formalize this intuition we generalize the $k$-phase model as follows.

**Definition 1.** *Let $I$ be a sequence and consider its $k$-decomposition. For constants $a > 1$ and $b < 1$, $I$ is called $(a,b)$-local if $L(I,k) \geq ak$ and each phase of $D(I,k)$ has at most $bk$ clean pages.*

Now we can define the fault rate of an algorithm $\mathcal{A}$ on $(a,b)$-local sequences, $F_{\mathcal{A}}(a,b)$, by restricting the input sequences to $(a,b)$-local sequences.

The following theorem shows that $\mathcal{RM}$ works better on sequences that are “more” local.

**Theorem 5.** *For any constants $a > 1$ and $b < 1$,*

$$F_{\mathcal{RM}}(a,b) \le \frac{b \cdot (H_k - H_{bk} + 1)}{a} \sim \frac{b(1-\ln b)}{a}.$$

*Proof.* Consider an arbitrary $(a,b)$-local sequence $I$ such that $|I| \ge n$. We should show that $F_{\mathcal{RM}}(I) \le \frac{b\cdot(H_k - H_{bk}+1)}{a}$. Consider the $k$-decomposition of $I$. Let $l_i$ denote the number of clean pages of phase $i$. Fiat et al. proved that the expected number of faults of $\mathcal{RM}$ in phase $i$, $f_i$, is at most $B_i = l_i \cdot (H_k - H_{l_i} + 1)$ [7]. Note that $1 \le l_i \le bk$; the first page of each phase is clean and $I$ is an $(a,b)$-local sequence. Since $B_i$ is strictly increasing for $l_i \le k$, we get $f_i \le b \cdot k \cdot (H_k - H_{bk} + 1)$. Therefore the expected number of faults that $\mathcal{RM}$ incurs on $I$ is at most $bk \cdot (H_k - H_{bk} + 1) \cdot |D(I,k)|$. On the other hand we have $|I| \ge ak \cdot |D(I,k)|$. Therefore

$$F_{\mathcal{RM}}(I) \le \frac{bk \cdot (H_k - H_{bk} + 1) \cdot |D(I,k)|}{ak \cdot |D(I,k)|} = \frac{b \cdot (H_k - H_{bk} + 1)}{a}.$$

□

Since $H_n \approx \ln n$, we can get an upper bound of $b \cdot (1 - \ln b)/a$ for $F_{\mathcal{RM}}(a,b)$. Thus the fault rate of $\mathcal{RM}$ decreases as $a$ increases and $b$ decreases. Note that several other results can be obtained by imposing more restrictions on input sequences. For example if $I$ is an $a$-local sequence that contains only $k+1$ distinct pages we have $l_i = 1$ for each phase $i$ and therefore

$$F_{\mathcal{RM}}(I) \le \frac{1 \cdot (H_k - H_1 + 1)|D(I,k)|}{ak|D(I,k)|} = \frac{H_k}{ak}.$$

## 6 Caching with Locality of Reference

In the paging problem, all pages have the same size and the same retrieval cost on a fault. However, in some applications such as caching files on the Web, pages have different sizes and the cost of bringing a page to the cache varies for different pages. We can generalize the paging problem in different ways. These generalized variants of the problem are usually called *caching* problems. There are various models for the caching problem [13, 8]:

- **General Model.** In this model pages can have arbitrary sizes and arbitrary retrieval costs.
- **Weighted Caching.** Pages have uniform sizes, but they can have arbitrary retrieval costs (weights).
- **Fault Model.** Pages have arbitrary weights, however, they have uniform retrieval costs.
- **Bit Model.** Pages have arbitrary sizes and the retrieval cost of a page is proportional to its size.

Each of these models is appropriate for certain applications. Irani describes some applications in Web caching that are best modeled using the Fault/Bit model [8]. In this section we study the behaviour of marking caching algorithms on sequences with high locality of reference.

### 6.1 Weighted Caching

For weighted caching, we introduce some new notation. Consider an on-line paging algorithm $\mathcal{A}$. Each page $\pi$ has a weight $w(\pi)$. In the full access cost model, the cost of a hit is 1 and the cost of a fault on a page $\pi$ is $p \cdot w(\pi) + 1$ for some parameter $p$. Let $W_{\mathcal{A}}(I)$ be the total weight of pages on which $\mathcal{A}$ incurs a fault when it serves a sequence $I$ and $W_{OPT}(I)$ be the same value for the optimal off-line algorithm. Define the average weight of faults in a phase as $AW_{\mathcal{A}}(I) = W_{\mathcal{A}}(I)/|D(I,k)|$ and $AW_{OPT}(I) = W_{OPT}(I)/|D(I,k)|$. Note that the full access cost of $\mathcal{A}$ and $OPT$ on $I$ is $|I| + p \cdot W_{\mathcal{A}}(I)$ and $|I| + p \cdot W_{OPT}(I)$, respectively. Let $C_{FA}(\mathcal{A}, I) = \frac{|I|+p \cdot W_{\mathcal{A}}(I)}{|I|+p \cdot W_{OPT}(I)}$; then we have $C_{FA}(\mathcal{A}) = \sup_I C_{FA}(\mathcal{A}, I)$.

Now assume that $I$ is an $a$-local sequence, i.e. $L(I,k) \geq ak$ for some constant $a > 1$. We have

$$C_{FA}(\mathcal{A}, I) = \frac{L(I,k) + p \cdot AW_{\mathcal{A}}(I)}{L(I,k) + p \cdot AW_{OPT}(I)} \leq \frac{ak + p \cdot AW_{\mathcal{A}}(I)}{ak + p \cdot AW_{OPT}(I)}.$$

Note that the standard competitive ratio of $\mathcal{A}$ is

$$C(\mathcal{A}) = \sup_I \frac{AW_{\mathcal{A}}(I)}{AW_{OPT}(I)}.$$

Therefore when $p$ is large, $C_{FA}(\mathcal{A})$ approaches the standard competitive ratio. For smaller values of $p$, $C_{FA}(\mathcal{A})$ improves as the locality of reference increases.

### 6.2 Bit Model

There is a close connection between the Bit model and the full access cost model. Let $s(\pi)$ denote the size of a page $\pi$ and $k$ be the size of cache. In the bit model, the retrieval cost of $\pi$ is $r \cdot s(\pi)$ for some fixed constant $r$. In the full access cost model, the cost of a hit is 1 and the cost of a fault is $p + 1$ for some parameter $p$. Therefore we can have a generalization of the full access cost model for the Bit model as follows. The cost of a hit is 1 and the cost of a fault on a page $\pi$ is $q \cdot s(\pi) + 1$ for some parameter $q$.

We obtain results in this model using the idea of $k$-decomposition. However since pages can have arbitrary sizes, we modify the definition of the decomposition. We upper bound the total size of distinct pages in a phase, rather than the number of distinct pages. For an input sequence $I$ and an integer $m > 1$, the $m$-decomposition $D(I,m)$ is defined as partitioning $I$ into consecutive phases so that each phase is a maximal subsequence that contains a set $\Pi$ of distinct pages such that the total of pages in $\Pi$ adds up to at most $m$ units of information.[1] Note that each phase may contain a set of distinct pages whose total size is strictly less than $m$. $|D(I,m)|$ and $L(I,m)$ are as before.

A marking algorithm in this model works in phases. At the beginning of each phase all pages in the cache are unmarked. A page is marked when it is

[1] Depending on the application, the unit of information can be: bit, byte, word, etc.

requested. On a fault, the algorithm brings the requested page to the cache and evicts as many (unmarked) pages as necessary from the cache to make room for this page. If all pages in the cache are marked, the phase ends and all pages are unmarked. As before, we call a sequence $I$ $a$-local if $L(I,k) \geq ak$ for some constant $a > 1$. Also assume that we have normalized the sizes of pages so that the smallest pages have unit size.

**Theorem 6.** *Let $\mathcal{A}$ be an arbitrary marking algorithm on an $a$-local input sequence. Then under the Bit model we have $C_{FA}(\mathcal{A}) \leq 1 + q/a$.*

*Proof.* Consider an arbitrary $a$-local sequence $I$. $\mathcal{A}$ incurs at most one fault on any page $\pi$ in any phase $\phi$ because $\pi$ is marked after the first fault and will not be evicted in the remaining steps of $\phi$. Since the total size of distinct pages in a phase is at most $k$, the full access cost of $\mathcal{A}$ on $I$ is at most $|I| + |D(I,k)| \cdot (q \cdot k)$. On the other hand, according to the definition of the decomposition, the optimal off-line algorithm should incur at lease one fault in each phase and therefore its full access cost is at least $|I| + |D(I,k)| \cdot (q \cdot 1)$. Therefore we get

$$C_{FA}(\mathcal{A}, I) \leq \frac{|I| + |D(I,k)| \cdot (q \cdot k)}{|I| + |D(I,k)| \cdot (q \cdot 1)} = \frac{L(I,k) + q \cdot k}{L(I,k) + q \cdot 1}.$$

Now since $I$ is an $a$local sequence, $L(I,k) \geq ak$ and

$$C_{FA}(\mathcal{A}, I) \leq \frac{ak + q \cdot k}{ak + q \cdot 1} = 1 + \frac{(k-1)}{ak/q + 1} < 1 + q/a.$$

This completes the proof as $I$ is an arbitrary $a$-local sequence. □

## 7 Conclusions

In this paper we studied some models for paging with locality of reference. In particular we proved that in general the competitive ratio does not improve on input sequences with high locality of reference under the models of Torng [12] and Albers et al. [1]. We also proposed a new model for locality of reference and proved that the randomized marking algorithm has better fault rate on sequences with high locality of reference. Finally we generalized the existing models to several variants of the caching problem.

*Acknowledgements* We thank Spyros Angelopoulos for useful discussions on this subject.

## References

1. S. Albers, L. M. Favrholdt, and O. Giel. On paging with locality of reference. *JCSS: Journal of Computer and System Sciences*, 70, 2005.

2. S. Angelopoulos, R. Dorrigiv, and A. López-Ortiz. On the separation and equivalence of paging strategies. In *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms (SODA '07)*, pages 229–237, 2007.
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. *Journal of Computer and System Sciences*, 50:244–258, 1995.
5. P. J. Denning. The working set model for program behaviour. *Communications of the ACM*, 11(5), May 1968.
6. R. Dorrigiv and A. López-Ortiz. A survey of performance measures for on-line algorithms. *SIGACTN: SIGACT News (ACM Special Interest Group on Automata and Computability Theory)*, 36(3):67–81, September 2005.
7. A. Fiat, R. M. Karp, M. Luby, L. A. McGeoch, D. D. Sleator, and N. E. Young. Competitive paging algorithms. *Journal of Algorithms*, 12(4):685–699, 1991.
8. S. Irani. Page replacement with multi-size pages and applications to web caching. *Algorithmica*, 33(3):384–409, 2002.
9. S. Irani, A. R. Karlin, and S. Phillips. Strongly competitive algorithms for paging with locality of reference. *SIAM J. Comput.*, 25:477–497, 1996.
10. K. Panagiotou and A. Souza. On adequate performance measures for paging. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC '06)*, pages 487–496, 2006.
11. D. D. Sleator and R. E. Tarjan. Amortized Efficiency of List Update and Paging Rules. *Communications of the ACM*, 28:202–208, 1985.
12. E. Torng. A unified analysis of paging and caching. *Algorithmica*, 20(2):175–200, 1998.
13. N. E. Young. On-line file caching. *Algorithmica*, 33(3):371–383, 2002.