BENJAMIN HILLER TJARK VREDEVELD

On the optimality of Least Recently Used

ZIB-Report 08-39 (September 2008)

On the optimality of Least Recently Used

Benjamin Hiller[*] Tjark Vredeveld[**]

September 29, 2008

Abstract

It is well known that competitive analysis yields too pessimistic results when applied to the paging problem and it also cannot make a distinction between many paging strategies. Many deterministic paging algorithms achieve the same competitive ratio, ranging from inefficient strategies as flush-when-full to the good performing least-recently-used (LRU).

In this paper, we study this fundamental online problem from the viewpoint of stochastic dominance. We show that when sequences are drawn from distributions modelling locality of reference, LRU is stochastically better than any other online paging algorithm.

1 Introduction

The *paging* problem is one of the most fundamental problems in online optimization. It models an optimization problem occuring in a two-level memory system. The first level, called the *slow memory*, stores a fixed set M of pages and the second level, which is the *fast memory* or *cache*, contains up to k pages of the set M. We will also refer to k as the *cache size*. In paging, one needs to serve a sequence of requests for pages $\sigma \in M^n$, where n is the number of pages requested. To serve a request for page $p \in M$, the system needs to have this page in the cache. If a requested page is not in the cache a *page fault* occurs. The requested page must then be loaded into the cache, and whenever the cache contains k pages, at least one page must be evicted from the fast memory. A paging algorithm needs to decide which page(s) will be evicted from the cache on a page fault. The goal is to minimize the number of page faults. Standard paging strategies include the following.

- *Least recently used* (LRU): whenever there is a page fault, evict the page whose most recent request was earliest.

[*]Zuse Institute Berlin, Takustraße 7, D-14195 Berlin, Germany, `hiller@zib.de`

[**]Maastricht University, Department of Quantitative Economics, P.O.Box 616, 6200 MD, Maastricht, The Netherlands, `t.vredeveld@ke.unimaas.nl`

- *First in first out* (FIFO): on a page fault, evict the page that has been in the cache the longest.
- *Deterministic marking algorithms*: whenever a page is requested, the page is labeled 'marked'. On a page fault and when the cache contains k pages, a marking algorithm evicts an 'unmarked' page from the cache. When there are no unmarked pages in the cache, it labels all pages as unmarked, before loading the (marked) page into the cache.
- *Flush when full* (FWF): On a page fault, when the cache is full, it evicts all pages from the cache.
- *Longest forward distance* (LFD): On a page fault, LFD evicts the page whose next request is farthest in the future.

All these algorithms, except LFD, are online algorithms. That is, they decide upon which page to evict without knowledge of which requests will come afterwards nor of the number of requests in the sequence. As LFD is not an online algorithm, it cannot be used in practice. However, LFD is an optimal page eviction strategy [5]. Note that LRU as well as FWF are both marking algorithms, whereas FIFO is not.

The standard yardstick for online algorithms has become *competitive analysis* [16, 13]. An online algorithm is called *c-competitive* if, for all request sequences, the cost of the algorithm, i. e., the number of page faults is at most c times the optimal offline costs. The smallest c for which an algorithm is c-competitive is also known as the *competitive ratio*. Sleator and Tarjan [16] showed that LRU and FIFO have a competitive ratio of k and that this is the best possible. Karlin et al. [13] gave a different proof for the same results and in addition they showed that FWF also has a competitive ratio of k. Torng [17] extended these results showing that all deterministic marking algorithms are k-competitive.

Related work. As FIFO, LRU and all other marking algorithms have the same competitive ratio, competitive analysis obviously fails to distinguish between these algorithms. Therefore, there has been research on the refinement of competitive analysis and alternative models for assessing online paging algorithms. Young [18, 20] introduced the notion of *loose competitiveness*, in which paging algorithms are evaluated for varying sizes of the fast memory, ignoring input sizes that have a high competitive ratio only for few cache sizes. He showed that several deterministic paging strategies where loosely $\mathcal{O}(\ln k)$ competitive. The Max/Max ratio of Ben-David and Borodin [6] compares the worst case amortized behavior of an algorithm with that of an optimal offline algorithm. An algorithm is said to have Max/Max ratio c if it is guaranteed that on no request sequence will it ever have to pay more than c times the maximal cost that an optimal offline algorithm pays on a sequence of the same length. Koutsoupias and Papadimitriou [14] introduced the diffuse adversary. In the concept of diffuse adversary, an average case competitive analysis is performed, but instead of selecting any probability distribution on the input sequence the diffuse adversary may only select a probability distribution from a prespecified class of distributions. Young [19] also performed a diffuse adversary analysis for the paging problem. They showed the optimality of LRU against some specific diffuse adversary. In the relative worst-order ratio [9] two algorithms are compared each on their respective worst-case permutation of a request sequence. Boyar et al. [9] showed that LRU is better than FWF, but LRU and FIFO are equally good according to this measure. Recenty, bijective analysis was introduced [2, 3]. In bijective analysis, one tries to find a bijective mapping from the set of instances on itself such that the preferred algorithm delivers on each instance a better objective function value than the algorithm, to which it is compared, has on the mapped instance. Angelopoulos and Schweitzer [3] showed that LRU is an optimal algorithm under this framework for a restricted class of sequences.

One of the reasons that competitive analysis is not able to make a distinction between the performance of several paging algorithms is that it considers arbitrary request sequences. In practice, however, request sequences have some structure, which is often refered to as *locality of reference*. This means that if a page is referenced, it is likely to be referenced again in the near future. Borodin et al. [8] presented the model of the *access graph* for locality of reference. The access graph models which pages can be requested after a certain page has been asked. Using this model, they showed that LRU is at least as good as FIFO.

Torng [17] introduces a model for locality of reference, based on Denning's working set concept [10, 11] by lower bounding the length of a subsequence containing a certain number of different pages. He shows that, among other algorithms, LRU achieves a constant competitive ratio.

Albers et al. [1] gave another model for locality of reference, also based on Denning's working set concept. They showed that LRU is an optimal online algorithm in their model and that FIFO and marking strategies are not optimal in general.

Becchetti [4] performs a diffuse adversary analysis, where the diffuse adversary is only allowed to choose a probability distribution on the request sequences that models locality of references. Given a certain prefix sequence, recently asked pages have a higher probability to be requested than not so recently asked pages. He shows that in this model, LRU outperforms FWF.

Our results. One of the weaknesses of competitive analysis is that it fails to distinguish between all kinds of algorithms. Therefore, alternative measures for the performance of online algorithms are needed. In this paper, we compare the performance of paging algorithms on random input sequences directly using stochastic dominance. This method for comparing online algorithms has been introduced in [12]. Given a probability distribution on all possible input sequences, we let X^{Alg} denote the random variable of the number of pages faults of an online algorithm Alg. We say that online algorithm Alg_1 is stochastically better than online algorithm Alg_2 if the random variable X^{Alg_1} is stochastically dominated by X^{Alg_2}, i. e., $\Pr\left[X^{\text{Alg}_1} \geq x\right] \leq \Pr\left[X^{\text{Alg}_2} \geq x\right]$ for all $x \in \mathbb{R}$.

In this paper, we discuss two families of distribution functions on the input sequences which both model locality of reference. The first family of probability distributions is due to Becchetti [4], who gives a higher probability to pages recently asked than those asked further in the past. The second family is a set of probability distributions inspired by the deterministic models for locality of reference introduced by Torng [17] and by Albers et al. [1].

For both families of distributions we give simple proofs that LRU is stochastically better than any other online algorithm. Moreover, we provide some ideas how to generalize these families maintaining the optimality of LRU. As the uniform distribution over all sequences of a certain length fits into the model of Becchetti, our results imply that also according to bijective analysis LRU is an optimal paging algorithm. Moreover, the optimality of LRU w. r. t. stochastic dominance implies that LRU is also a best algorithm w. r. t. the average number of page faults as well as the average competitive ratio.

2 Mode of analysis and locality of reference models

We start by recalling some basic notions for paging algorithms. A standard tool is the partitioning of a sequence into *phases*, see e. g., [7]. The first phase starts with the first request. Phase ℓ starts with the $(k+1)$st *distinct* request after the start of phase $\ell - 1$. Each phase ends just before the start of the next phase or at the end of the sequence, whichever comes first.

Given a request sequence σ, we say that a page p is *marked w. r. t. σ* if it has been requested in the final phase of σ; otherwise, we say that p is *unmarked w. r. t. σ*. Note that by definition of the phases, there cannot be more than k marked pages w. r. t. a request sequence. Also note that all pages are unmarked w. r. t. the empty sequence. Moreover, the partition into phases and the set of marked pages at any point in the sequence do not depend on the algorithm. Observe that a marking algorithm has, at any point in time, all marked pages in its cache, which justifies the name.

2.1 Stochastic dominance analysis of online algorithms

The competitive ratio as a measure of the performance of an online algorithm has been critized for being too pessimistic. For the paging problem, it fails to discriminate between algorithms that perform very differently in practice.

In our approach [12] we compare the performance of algorithms on random request sequences drawn according to certain probability distributions. In contrast to competitive analysis or diffuse adversary analysis, we directly compare two algorithms to each other without refering to an optimal offline solution. We compare the performance of online algorithms using *stochastic dominance*, a well-known stochastic order. A random variable X is said to be *stochastically dominated* by a random variable Y, written $X \leq_{\text{st}} Y$, if

$$\Pr[X \geq x] \leq \Pr[Y \geq x] \quad \text{for all } x \in \mathbb{R}. \tag{1}$$

One way to think of this approach is that we compare the *distributions* of the performances of two online algorithms instead of aggregate statistics like the expected value or the maximum. We will later see that in some cases, there are distribution-free interpretations of a stochastic dominance result.

Stochastic dominance has some interesting properties [15]. Abusing notation, we denote the random variable for the performance of an algorithm Alg by the same symbol, Alg. The first interesting consequence of stochastic domination of the performance of one algorithm by another, i. e., $\text{Alg}_1 \leq_{\text{st}} \text{Alg}_2$, is that the expected performance of Alg_1 is also better than that of Alg_2, i. e., $\mathbb{E}[\text{Alg}_1] \leq \mathbb{E}[\text{Alg}_2]$. This again implies that the average competitive ratio of Alg_1 is not worse than that of Alg_2: $\mathbb{E}[\text{Alg}_1]/\mathbb{E}[\text{Opt}] \leq \mathbb{E}[\text{Alg}_2]/\mathbb{E}[\text{Opt}]$. Finally, if we have an increasing function f on the possible outcomes for Alg_1 and Alg_2, then $f(\text{Alg}_1) \leq_{\text{st}} f(\text{Alg}_2)$. This can be used to conclude that Alg_1 is also better than Alg_2 in the full access cost model [17], where a page in the cache incurs cost 1, when requested, and a page fault incurs cost $1 + p$ for some parameter $p > 0$.

In case $\text{Alg}_1 \leq_{\text{st}} \text{Alg}_2$ holds for the uniform distribution on (a subset of) the sequences, this is equivalent to the existence of a bijective mapping ϕ on the sequences such that $\text{Alg}_1(\sigma) \leq \text{Alg}_2(\phi(\sigma))$ for any σ. This strong way of comparing online algorithms is called *bijective analysis* and has been introduced in [2]. Thus stochastic dominance results for the uniform distribution share the favorable properties of bijective analysis results discussed in [2].

2.2 Paging with locality of reference

There are three models for paging with locality of reference we are aware of [17, 4, 1]. We define families of probability distributions on the request sequences based on these models. Like in [14, 19, 4], the considered probability distributions on the request sequences are completely described by the probability that page p is requested given that the sequence up to this page is σ, $\Pr[p \mid \sigma]$.

The age model Becchetti [4] introduced the following probabilistic model which we call *age model.* In the age model, the next request for a prefix sequence σ is generated based on the age of the pages. For a prefix sequence σ, the age of a page $p \in M$ is defined by

$$\operatorname{age}(p, \sigma) := \begin{cases} l & \text{if } p \text{ is the } l\text{th most recently requested page,} \\ \infty & \text{if } p \text{ does not appear in } \sigma. \end{cases}$$

We say that a probability distribution over the request sequences is an *age model distribution* if it arises in the following way. Let $\mathcal{D}$ be the set of distributions over $\{1, \ldots, |M|\}$ with monotone non-increasing distribution functions. Given a prefix sequence σ, the probability $\Pr[p \mid \sigma]$ is determined by an *age distribution* $\delta \in \mathcal{D}$. The age distribution δ gives the age of the new request page p, i. e., if a is a realization according to δ, the next page is $p \in M$ with $\operatorname{age}(p, \sigma) = a$. If there is no page with age a one of the pages with age ∞ is chosen arbitrarily.

Note that considering age distributions from $\mathcal{D}$ models locality of reference: Pages requested more recently have a high probability to be requested next. Becchetti [4] has additional restrictions on the distributions in $\mathcal{D}$, but we will only consider this more general probabilistic model for locality of reference.

The concave function model and the k-phase model In contrast to the age model, the concave function model [1] and the k-phase model [17] are deterministic models which restrict the set of request sequences.

Albers et al. [1] propose the *concave function model* which models working sets. Locality of reference is modeled by a concave function $f\colon \mathbb{N} \to \mathbb{N}$, which specifies the maximum number $f(l)$ of distinct pages in a (contiguous) subsequence of length l for any $l \in \mathbb{N}$. A request sequence for which each subsequence of length l has at most $f(l)$ distinct pages is called *f-consistent.*

For a fixed concave function $f\colon \mathbb{N} \to \mathbb{N}$, we consider the following probability distribution on f-consistent sequences which we call the *f-consistent distribution.* The idea is that for a prefix sequence σ, the next request is chosen uniformly at random such that the resulting sequence is still f-consistent. Hereto, we denote by $\lambda(\sigma, l)$ the subsequence consisting of the last l requests of σ and $M(\sigma')$ is the set of distinct pages in a sequence σ'. Furthermore, $l^*(\sigma)$ is the smallest l such that $|M(\lambda(\sigma, l))| = f(l)$, which is ∞ if this condition is not satisfied for any l. Finally, we denote the pages requested in the last $l^*(\sigma)$ requests by $M^*(\sigma)$, i. e., $M^*(\sigma) = M(\lambda(\sigma, l^*(\sigma)))$, for $l^*(\sigma) < \infty$, and $M^*(\sigma) = M(\sigma)$ otherwise.

Suppose the sequence σ has already been generated. The next request is chosen

- uniformly at random from M if $l^*(\sigma) = \infty$ or

- uniformly at random from $M^*(\sigma)$ if $l^*(\sigma) < \infty$.

Proposition 2.1 *Every sequence which has a positive probability under the f-consistent distribution is f-consistent.*

Proof. Suppose σ is generated as described above. Since every 1-element sequence is trivially f-consistent, we can assume for purposes of induction that $\sigma' = (\sigma, p)$ with σ being f-consistent. By definition, all subsequences of σ are f-consistent. It is therefore sufficient to show that $\lambda(\sigma', l)$ satisfies $|M(\lambda(\sigma', l))| \leq f(l)$ for $2 \leq l \leq |\sigma'|$. If $l < l^*(\sigma)$ we have that

$$|M(\lambda(\sigma', l))| \leq |M(\lambda(\sigma, l-1))| + 1 \leq f(l),$$

whereas in the case $l \geq l^*(\sigma)$

$$|M(\lambda(\sigma', l))| = |M(\lambda(\sigma, l-1))| \leq f(l)$$

holds. □

Torng [17] introduces the *k-phase model* for locality of reference by using the partitioning of each sequence into phases and requiring that each phase has a minimum length. In particular, a sequence σ is called *a-local* for some $a \geq 1$ if every phase of σ consists of at least ak requests.

Similarly to the concave function model, we consider the following probability distribution on the a-local sequences which we call *a-local distribution.* Suppose the prefix sequence σ has already been generated and denote by $\rho(\sigma)$ the subsequence of σ corresponding to the last phase and by $M(\rho(\sigma))$ the set of the corresponding pages. The next request is chosen

- uniformly at random from $M(\rho(\sigma))$ if $|M(\rho(\sigma))| = k$ and $|\rho(\sigma)| < ak$ or
- uniformly at random from M otherwise.

Obviously, the generated sequences are a-local.

3 Optimality of LRU for paging with locality of reference

In this section we show that LRU incurs stochastically fewer page faults than any other paging algorithm for paging with locality of reference, i. e., LRU is optimal w. r. t. to stochastic dominance. In particular, we show that LRU is an optimal algorithm for the probability distributions over the sequences defined in Section 2.2. We also note that LRU is also optimal for natural probability distributions over a larger class of sequences exhibiting locality of reference.

3.1 Preliminaries

Given an online algorithm Alg and a prefix sequence σ, the random variable $W^{\text{Alg}}(\sigma) = 1$ if the next request after σ leads to a page fault in Alg, and $W^{\text{Alg}}(\sigma) = 0$ otherwise. For given $j = 1, \ldots, n$ and a sequence σ of length $|\sigma| = j - 1$, the random variable $X_j^{\text{Alg}}(\sigma) = 1$ if the first $j - 1$ requests are given by σ and Alg encounters a page fault on the jth request; otherwise $X_j^{\text{Alg}}(\sigma) = 1$. The random variable $Y_j(\sigma) = 1$ if $|\sigma| = j$ and the first j requests are as in σ; otherwise $Y_j(\sigma) = 0$. Given a sequence σ of length $|\sigma| \geq j$ and an online algorithm Alg, the variable $Z_j^{\text{Alg}}(\sigma) = 1$ if the jth request leads to a page fault when Alg operates on σ and $Z_j^{\text{Alg}}(\sigma) = 0$ otherwise. Note that $Z_j^{\text{Alg}}(\sigma)$ is deterministically determined by Alg, j, and σ.

On a sequence $\sigma \in M^n$, an online algorithm Alg has $\sum_{j=1}^{n} Z_j^{\text{Alg}}(\sigma)$ page faults. Therefore, the value of Alg can be expressed as

$$\text{Alg} = \sum_{\sigma \in M^n} \sum_{j=1}^{n} Z_j^{\text{Alg}}(\sigma) Y_n(\sigma).$$

Finally, we denote by $C^{\text{Alg}}(\sigma)$ the of pages in the cache if the sequence σ is processed by algorithm Alg.

Lemma 3.1 *The value of an online algorithm* Alg *can be written as*

$$\text{Alg} = \sum_{j=1}^{n} \sum_{\sigma \in M^{j-1}} X_j^{\text{Alg}}(\sigma).$$

Proof.

$$\begin{aligned}\text{Alg} &= \sum_{\sigma \in M^n} \sum_{j=1}^{n} Z_j^{\text{Alg}}(\sigma) Y_n(\sigma) = \sum_{j=1}^{n} \sum_{\sigma \in M^n} Z_j^{\text{Alg}}(\sigma) Y_n(\sigma) \\ &= \sum_{j=1}^{n} \sum_{\sigma_1 \in M^{j-1}} \sum_{p \in M} \sum_{\sigma_2 \in M^{n-j}} Z_j^{\text{Alg}}(\sigma_1 p \sigma_2) Y_n(\sigma_1 p \sigma_2)\end{aligned}$$

as Alg is an online algorithm, we have

$$= \sum_{j=1}^{n} \sum_{\sigma_1 \in M^{j-1}} \sum_{p \in M} Z_j^{\text{Alg}}(\sigma_1 p) \sum_{\sigma_2 \in M^{n-j}} Y_n(\sigma_1 p \sigma_2)$$

due to the fact that $Y_j(\sigma_1) = \sum_{\sigma_2 \in M^{n-j}} Y_n(\sigma_1 \sigma_2)$, we can write

$$\begin{aligned}&= \sum_{j=1}^{n} \sum_{\sigma_1 \in M^{j-1}} \sum_{p \in M} Z_j^{\text{Alg}}(\sigma_1 p) Y_j(\sigma_1 p) \\ &= \sum_{j=1}^{n} \sum_{\sigma_1 \in M^{j-1}} X_j^{\text{Alg}}(\sigma_1),\end{aligned}$$

where the last equality follows from that fact that $X_j^{\text{Alg}}(\sigma_1) = 1$ for all realizations of the request sequence that start with σ_1 and the jth request p leads to a page fault, i. e., $Z_j^{\text{Alg}}(\sigma_1 p) = 1$. $\square$

Lemma 3.2 *Let* Alg_1 *and* Alg_2 *be two online paging algorithms. Denote by* P *the random request generated after prefix sequence* σ*. Suppose that*

$$\Pr\left[P \in C^{\text{Alg}_1}(\sigma)\right] \geq \Pr\left[P \in C^{\text{Alg}_2}(\sigma)\right] \tag{2}$$

for any sequence σ*. Then* $\text{Alg}_1 \leq_{\text{st}} \text{Alg}_2$.

Proof. By Lemma 3.1, we only need to show

$$X_j^{\text{Alg}_1}(\sigma) \leq_{\text{st}} X_j^{\text{Alg}_2}(\sigma),$$

for all $j = 1, \ldots, n$, and all sequences $\sigma \in M^{j-1}$. As the variables $X_j^{\text{Alg}}(\sigma)$ are binary random variables, this is equivalent to

$$\Pr\left[X_j^{\text{Alg}_1}(\sigma) = 1\right] \leq \Pr\left[X_j^{\text{Alg}_2}(\sigma) = 1\right]. \tag{3}$$

For any online algorithm Alg, $j \in \{1, \ldots, n\}$, and $\sigma \in M^{j-1}$, we can write

$$\begin{aligned}
\Pr\left[X_j^{\text{Alg}}(\sigma) = 1\right] &= \Pr\left[Y_{j-1}(\sigma) = 1 \wedge X_j^{\text{Alg}}(\sigma) = 1\right] \\
&= \Pr\left[X_j^{\text{Alg}}(\sigma) = 1 \,|\, Y_{j-1}(\sigma) = 1\right] \cdot \Pr\left[Y_{j-1}(\sigma) = 1\right] \\
&= \Pr\left[W^{\text{Alg}}(\sigma) = 1 \,|\, Y_{j-1}(\sigma) = 1\right] \cdot \Pr\left[Y_{j-1}(\sigma) = 1\right] \\
&= \Pr\left[W^{\text{Alg}}(\sigma) = 1\right] \cdot \Pr\left[Y_{j-1}(\sigma) = 1\right].
\end{aligned}$$

Since $Y_{j-1}(\sigma)$ does not depend on the algorithm, (2) is equivalent to $\Pr\left[W^{\text{Alg}_1}(\sigma)\right] \leq \Pr\left[W^{\text{Alg}_2}(\sigma)\right]$ and thus it implies (3), $\square$

3.2 Optimality results

Theorem 3.3 *Suppose the request sequence is chosen according to an age model distribution. Then the number of page faults of LRU is stochastically dominated by that of any online algorithm.*

Proof. Let Alg be any online paging algorithm. We show that condition (2) of Lemma 3.2 is satisfied. Note that always $|C^{\text{LRU}}(\sigma)| \geq |C^{\text{Alg}}(\sigma)|$ holds. By definition of LRU, there is an injective mapping $\phi\colon C^{\text{Alg}}(\sigma) \to C^{\text{LRU}}(\sigma)$ that maps a page from $C^{\text{Alg}}(\sigma)$ to a page in $C^{\text{LRU}}(\sigma)$ which is not older. Let p be some page that Alg has in the cache and denote by P the random next request generated according to the age model distribution. Clearly, $\Pr\left[P = \phi(p)\right] \geq \Pr\left[P = p\right]$, which implies condition (2). $\square$

As the uniform distribution over all sequences belongs to the family of age model distributions, we have the following corollary to the previous theorem.

Corollary 3.4 *(1) LRU is an optimal algorithm w. r. t. bijective analysis for the set of all request sequences. (2) LRU has the best average competitive ratio when the request sequence is chosen from an age model distribution.*

Theorem 3.5 *The number of page faults of LRU is stochastically dominated by that of any online algorithm when the request sequence is chosen*

1. *according to the f-consistent distribution or*
2. *according to the a-local distribution.*

Proof. Let us first address the f-consistent distribution. Consider the prefix sequence σ. In the case that $|M^*(\sigma)| \leq k$, LRU has all pages in cache that have been asked during the last $l^*(\sigma)$ page requests. Therefore,

$$\Pr\left[p \in C^{\mathrm{LRU}}(\sigma)\right] = 1 \geq \Pr\left[p \in C^{\mathrm{Alg}}(\sigma)\right].$$

On the other hand, if $|M^*(\sigma)| \geq k$, we have

$$\Pr\left[p \in C^{\mathrm{LRU}}(\sigma)\right] = \frac{k}{|M^*(\sigma)|} \geq \Pr\left[p \in C^{\mathrm{Alg}}(\sigma)\right],$$

as any algorithm can have at most k pages in its cache and LRU has the k most recently asked pages, which is a subset of $M^*(\sigma)$, in its fast memory.

The proof for the a-local distribution is similar. $\square$

Note that both the f-consistent and the a-local distribution are *not* the uniform distributions on the respective sequence sets. Therefore Theorem 3.5 does not imply bijective analysis results.

Remark In fact, the proof of Theorem 3.5 works for a larger class of locality of reference models. For a given sequence σ, we define $N(\sigma)$ as the set of all possible pages that may be asked as the next request. For example, for the f-consistent distribution $N(\sigma) = M^*(\sigma)$ or $N(\sigma) = M$. For the f-consistent and a-local distributions, the set $N(\sigma)$ consists of the most recent requests or it is equal to M. We relax this condition in such a way that if there are only few, i. e., at most k, possible pages to be asked, then $N(\sigma)$ should be contained in the set of k most recent pages. Otherwise, of course, the k most recent pages should be contained in $N(\sigma)$. Like in the f-consistent and the a-local distributions, request sequences arise by choosing the next request uniformly at random from the set $N(\sigma)$, given a sequence σ seen so far. These conditions on $N(\sigma)$ are essentially the arguments used in the proof of Theorem 3.5. Therefore, LRU is also stochastically optimal in this model.

References

[1] S. Albers, L. M. Favrholdt, and O. Giel. On paging with locality of reference. *Journal on Computer and System Sciences*, 70(2):145–175, 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*, pages 229–237, 2007.

[3] S. Angelopoulos and P. Schweitzer. Paging and list update under bijective analysis. In *Proceedings of the 20th ACM-SIAM symposium on Discrete algorithms*, 2009. To appear.

[4] L. Becchetti. Modeling locality: A probabilistic analysis of LRU and FWF. In *Proceedings of the 12th European Symp. on Algorithms (ESA)*, pages 98–109, 2004.

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

[6] S. Ben-David and A. Borodin. A new measure for the study of online algorithms. *Algorithmica*, 11(1):73–91, 1994.

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

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

[9] J. Boyar, L. M. Favrholdt, and K. S. Larsen. The relative worst-order ratio applied to paging. *Journal on Computer and System Sciences*, 73(6):818–843, 2007.

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

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

[12] B. Hiller and T. Vredeveld. Probabilistic analysis of online bin coloring algorithms via stochastic comparison. In *Proceedings of the 16th Annual European Symposium on Algorithms*, Lecture Notes in Computer Science, 2008. to appear.

[13] A. Karlin, M. Manasse, L. Rudolph, and D. Sleator. Competitive snoopy paging. *Algorithmica*, 3:79–119, 1988.

[14] E. Koutsoupias and C. H. Papadimitriou. Beyond competitive analysis. *SIAM Journal on Computing*, 30(1):300–317, 2000.

[15] A. Müller and D. Stoyan. *Comparison Models for Stochastic Models and Risks*. John Wiley & Sons, 2002.

[16] D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.

[17] E. Torng. A unified analysis of paging and caching. *Algorithmica*, 20(1):175–200, 1998.

[18] N. E. Young. The k-server dual and loose competitiveness for paging. *Algorithmica*, 11(6):525–541, 1994.

[19] N. E. Young. On-line paging against adversarially biased random inputs. *Journal of Algorithms*, 37(1):218–235, 2000.

[20] N. E. Young. On-line file caching. *Algorithmica*, 33(3):371–383, 2002.