Bounding the Diffuse Adversary

Neal E. Young*

Abstract

Koutsoupias and Papadimitriou recently raised the question of how well deterministic on-line paging algorithms can do against a certain class of *adversarially biased* random inputs [3]. Such an input is given in an on-line fashion; the adversary determines the next request probabilistically, subject to the constraint that no page may be requested with probability *more than* a fixed $\epsilon > 0$.

In this note, we answer their question by estimating, within a factor of two, the optimal competitive ratio of any deterministic on-line strategy against this adversary. We further analyze randomized on-line strategies, obtaining upper and lower bounds within a factor of two. These estimates reveal the qualitative changes as ϵ ranges continuously from 1 (the standard model) towards 0 (a severely handicapped adversary). Our upper bounds use an "insurance-based" charging scheme that reduces the analysis of the expected cost on an adversarially biased random input to the analysis of the "adjusted" cost of a worst-case input.

1 Introduction

Measuring an algorithm by its theoretical worst-case performance is often impractically pessimistic. On the other hand, measuring an algorithm by its average-case performance on a specific input distribution may be impractically presumptuous. An approach between these two extremes is to assume that *something*, but not *everything*, is known about the input distribution — namely, that the input for an algorithm has been generated by a random source that has been somehow *adversarially biased* by an adversary that chooses the worst possible bias for the algorithm in question.

Koutsoupias and Papadimitriou [3] recently studied the performance of deterministic on-line algorithms against such an adversary. Their adversary, Δ_ϵ, is allowed to select the next request only probabilistically, with each page being requested with probability at most some $\epsilon > 0$. Koutsoupias and Papadimitriou prove that the least-recently-used strategy (LRU) is an optimal deterministic on-line algorithm against this adversary, but they leave open the problem of giving a closed-form estimate of the optimal competitive ratio $R(\Delta_\epsilon)$, commenting "It seems difficult to determine ... the exact competitive ratio. ... In fact ... the ratio may not be expressible as a simple closed form expression." In this note, we estimate this ratio within a factor of roughly two for all k and ϵ.

*Dartmouth College, Hanover NH 03755 ney@dartmouth.edu

1.1 Results

Define $\Phi(\epsilon, k) \doteq 1 + \sum_{i=1}^{k-1} 1/\max\{\epsilon^{-1} - i, 1\}$.

THEOREM 1.1. *Fix any integer $k > 0$ and $\epsilon \in (0, 1]$.*

The competitive ratio of any deterministic on-line algorithm versus the diffuse adversary Δ_ϵ is at least $\Phi(\epsilon, k) - 1$.

Conversely, the ratio for any deterministic "marking" algorithm (including LRU *and first-in-first-out (*FIFO*)) is at most $2\Phi(\epsilon, k)$.*

The upper bound does not hold for flush-when-full (FWF). For *randomized* strategies, the optimal competitive ratio exhibits a curious behavior. For ϵ below the threshold $1/k$, randomized strategies aren't much better than *deterministic ones*: For ϵ above this threshold, randomized strategies don't do much better than *against the standard adversary.*

THEOREM 1.2. *Fix any integer $k > 0$ and any $\epsilon > 0$.*

*If ϵ^{-1} is an integer greater than k, the lower bound in Theorem 1.1 also applies to any randomized on-line algorithm. Conversely, the upper bound there also applies to the randomized marking algorithm (*MARK*).*

For $\epsilon \geq 1/(k+1)$, the competitive ratio of any randomized on-line algorithm versus the diffuse adversary Δ_ϵ is at least $\Phi(1/(k+1), k) = H(k) - 1$. Conversely, the ratio for MARK *is at most $2H(k)$.*

(The last paragraph of the above Theorem essentially follows from known results about MARK [2, 6, 7].) The above results are summarized in Table 1. Estimates of Φ are given in Table 2. Note: $H(k) = \sum_{i=1}^{k} \frac{1}{i} \approx \ln(k+1)$.

That all marking algorithms are within a constant factor of optimal might be considered a failing of the diffuse-adversary model — in practice, LRU and its variants are generally considered to be better than FIFO.

Our upper bounds use a charging scheme for analyzing adversarially biased random inputs. The charging scheme adjusts the costs of each input in such a way that the expected cost of a random input is unchanged, but so that, working with the adjusted costs, we can show *worst-case* bounds on each input. This allows us to reduce the analysis to a worst-case analysis, while still thinking of some of the costs as expected costs.

Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, 420-425 (1998)

range	deterministic lower bound	deterministic upper bound	randomized lower bound	randomized upper bound
$\epsilon \leq 1/k$	$\Phi(\epsilon, k) - 1$	$2\Phi(\epsilon, k)$	$\Phi(\epsilon', k) - 1$	$2\Phi(\epsilon, k)$
$\epsilon \geq 1/k$			$H(k) - 1$	$2H(k)$

Table 1: Upper and lower bounds on the optimal competitive ratio of on-line algorithms with cache size k against Koutsoupias and Papadimitriou's diffuse adversary Δ_ϵ. The upper bounds hold for any deterministic marking algorithm (e.g. LRU, FIFO) and for the randomized marking algorithm MARK. In the randomized lower bound, $\epsilon' = 1/\lceil 1/\epsilon \rceil$. Note: $H(k) = \sum_1^k 1/i = \Phi(1/(k+1), k) \approx \ln(k+1)$.

range	exact value of $\Phi(\epsilon, k)$	approximation
$\epsilon \leq 1/k$	$1 + \sum_{i=1}^{k-1} \frac{1}{\epsilon^{-1} - i}$	$1 + \ln \frac{1}{1 - \epsilon k + \epsilon}$
$\epsilon \geq 1/k$	$1 + k - \lfloor \epsilon^{-1} \rfloor + \sum_{i=1}^{\lfloor \epsilon^{-1} \rfloor - 1} \frac{1}{\epsilon^{-1} - i}$	$1 + k - \epsilon^{-1} + \ln \epsilon^{-1}$

Table 2: Exact and approximate values of $\Phi(\epsilon, k)$ (see Theorems 1.1 and 1.2). As ϵ varies from 0 to 1, $\Phi(\epsilon, k)$ varies from constant to linear in k. Around the threshold $\epsilon \approx 1/k$, $\Phi(\epsilon, k)$ is logarithmic in k.

1.2 Definitions

The *paging problem* [5], given an integer $k > 0$, is to dynamically maintain a cache (set) of at most k pages in response to a sequence of requests for pages so as to minimize the number of *page faults*. A page fault occurs when the requested page is not in the cache, at which point the page must be brought into the cache. If there are k pages in the cache already, one must be evicted (removed) before the requested page is brought in. An algorithm for the problem must specify which page to evict when a fault occurs. Given an algorithm A and a sequence x, we let $A(x)$ denote the cost (number of faults) incurred by A in servicing x. If A is a randomized algorithm, then $A(x)$ denotes the expected cost (over random choices made by the algorithm) on input x. The optimal algorithm, OPT [1], evicts the page that will be next requested latest. An algorithm is *on-line* if the choice of which page to evict is independent of subsequent requests.

On-line paging strategies considered here include the following. *Least-recently-used* (LRU) evicts the page whose most recent request is the least recent among all pages in the cache. *First-in-first-out* (FIFO) evicts the page that has been in the cache the longest. The *randomized marking algorithm* (MARK) operates as follows. After a page is requested, it is marked. When a page is to be evicted, an unmarked page is chosen uniformly at random, with the caveat that if all pages in the cache are marked, then all marks are first erased. The phrase *deterministic marking algorithm* (DMARK) refers to any deterministic algorithm that maintains marks as MARK does and evicts only unmarked pages. LRU and FIFO are examples.

Following Koutsoupias and Papadimitriou [3], given a known class of distributions Δ of the input sequences, and an algorithm A, define

$$R(\Delta, A) \doteq \max_{D \in \Delta} \frac{\mathrm{E}_D[A(x)]}{\mathrm{E}_D[\mathrm{OPT}(x)]}$$

and

$$R(\Delta) \doteq \min_A R(\Delta, A),$$

where A ranges over all deterministic on-line algorithms, and

$$R_r(\Delta) = \min_A R(\Delta, A),$$

where A ranges over all randomized on-line algorithms. The parameter k is implicit in these definitions.) Koutsoupias and Papadimitriou call this the *diffuse adversary* model.

Any distribution D specifies, for each page p and sequence of page requests x, the probability $\Pr_D(p|x)$ that the next request is p given that the sequence so far is x. Define Δ_ϵ to contain those distributions D such that, for any request sequence x and page p, $\Pr_D(p|x) \leq \epsilon$. We are interested in estimating $R(\Delta_\epsilon)$ and $R_r(\Delta_\epsilon)$.

2 Upper Bound on Deterministic Strategies

Let DMARK denote any deterministic marking algorithm. (The reader may wish to consider the example of LRU for concreteness.) Fix any distribution $D \in \Delta_\epsilon$. Below, by "a random sequence x", we mean x is chosen randomly according to distribution D.

For any sequence x, let $\overline{\text{DMARK}}(x)$ denote an "adjusted" cost (defined below) of running DMARK on x. The key properties of the adjusted cost are:

1. $\mathrm{E}_D[\text{DMARK}(x)] \leq \mathrm{E}_D[\overline{\text{DMARK}}(x)]$ for *random* x.
2. $\overline{\text{DMARK}}(x) \leq c \cdot \text{OPT}(x)$ for *any* x (for some c).

These give the desired upper bound via

$$\begin{aligned}\frac{\mathrm{E}[\text{DMARK}(x)]}{\mathrm{E}[\text{OPT}(x)]} &\leq \frac{\mathrm{E}[\overline{\text{DMARK}}(x)]}{\mathrm{E}[\text{OPT}(x)]} \\ &\leq \max_x \frac{\overline{\text{DMARK}}(x)}{\text{OPT}(x)} \\ &\leq c.\end{aligned}$$

2.1 Adjusted Cost: Terms and Motivation

We now motivate the adjusted cost $\overline{\text{DMARK}}(x)$. Fix any sequence x. Partition x into a sequence of contiguous subsequences, called *phases*, as follows. The first phase starts with the first request. In general, when each phase starts, DMARK has k unmarked pages in its cache. As requests occur, DMARK evicts unmarked pages (as necessary) and brings in requested pages (as necessary) and marks them. Once a page is requested, it remains marked and in the cache for the rest of the phase. The phase ends when all pages remaining in the cache are marked and the next request is not to one of those pages. This causes DMARK to unmark the pages.

This action — unmarking the pages — begins the next phase. Thus, DMARK starts each phase with the k pages requested in the previous phase unmarked in the cache. It immediately evicts one of them in response to the first request of the phase.

Within each phase, classify the requests as follows:

New — a request to a page that was not requested the previous phase. The first request of the phase is always new.

Worrisome — a request to a page that was requested in the previous phase, but nonetheless causes DMARK to fault (because DMARK has evicted the page previously during the phase).

Redundant — a request to an already marked page.

New and worrisome requests are the only requests that cause DMARK to fault.

It was previously observed [2, 6, 7] that in a phase with m new requests, OPT incurs at least $m/2$ faults (amortized over the sequence). Briefly, this is because in two consecutive phases, if the second has m new requests, then $k+m$ distinct pages are requested. Since OPT has a cache of size k, it must incur at least m faults during the two phases.

On the other hand, OPT incurs at most m faults in the phase (again amortized over the sequence). Briefly, this is because OPT has the option of starting the phase with the k pages from last phase and then evicting just m of these pages — those that won't be requested this phase.

Thus, the new requests are not a problem — OPT is also paying for those. The worrisome requests are the problem — they are the only other requests that cause DMARK to fault. We want to use the phase structure of a sequence to give a competitive analysis. On the other hand, we can only hope to bound the cost of the worrisome requests in the *expected* sense for a *random* sequence — this is where our analysis needs to take into account the limitations of the diffuse adversary. We cannot condition the random sequence on having a particular phase structure — the limitation on the diffuse adversary does not extend to *conditioned* random inputs.

2.2 Adjusted Cost: Definition

Instead, we work with adjusted costs, as mentioned above. On any sequence x, define $\overline{\text{DMARK}}(x)$ to be the number of new requests in x plus the total of the *insurance premiums* for x. There is an insurance premium for each non-redundant request p in x. The premium equals the probability that the next non-redundant request would be a worrisome request *if the remaining requests (subsequent to p) were generated randomly according to the distribution D.* In effect, $\overline{\text{DMARK}}$ pays directly for new requests, but instead of paying for worrisome requests, it buys "insurance" in advance. Because the cost of the insurance equals the expected savings when the next non-redundant request occurs, we have:

LEMMA 2.1. *For any* $D \in \Delta_\epsilon$,

$$\mathrm{E}_D[\text{DMARK}(x)] \leq \mathrm{E}_D\left[\overline{\text{DMARK}}(x)\right].$$

2.3 Worst-Case Bound

We have now reduced the problem to giving a worst-case bound on the *adjusted* cost of any sequence in terms of the cost OPT incurs on that sequence. Fix any sequence x and consider a phase within the sequence. Let m denote the number of new requests in the phase. The main task is bounding the cost of the insurance premiums paid during the phase; the only other cost charged to DMARK is m for new requests, for which OPT also pays $m/2$. We bound the cost of the premiums in terms of m.

Insurance premiums are paid following each of the k non-redundant requests in the phase. Let p be any non-redundant request in the phase. Consider the requests in the phase up to and including p. Let i be the number of non-redundant requests so far ($1 \leq i \leq k$; for the first request in the phase, $i = 1$).

There are at most m pages that, if requested next, would result in a worrisome request (of the k pages requested in the previous phase, at most m are not in the cache). There are at most i pages that, if requested next, would result in a redundant request (the i distinct pages requested so far). If any other page were requested next, it would either be a new request or a non-worrisome, non-redundant request (to a page in DMARK's cache). Thus, the only way that the next non-redundant request could be a worrisome request is if the upcoming sequence of requests were to consist of some sequence of the i possible redundant requests followed by a request to one of the at most m possible worrisome requests.

Since the adversary can assign a probability of at most ϵ to any page, the probability that the next non-redundant request would be worrisome is bounded by

$$\sum_{\ell \geq 0} (\epsilon i)^{\ell} \epsilon m = \frac{\epsilon m}{1 - \epsilon i} = \frac{m}{\epsilon^{-1} - i}$$

(or 1 if the quantity on the right-hand side is negative or more than 1). Above ℓ is the number of redundant requests before the next non-redundant request. This is our upper bound on the insurance payment paid by DMARK after request p. Summing all the payments and dividing by $m/2$ (our lower bound on the amortized cost of OPT), and appealing to Lemma 2.1 gives:

LEMMA 2.2. *For any diffuse adversary $D \in \Delta_\epsilon$, the expected cost of any deterministic marking algorithm* DMARK *on a random sequence from D, divided by the expected cost of* OPT *on a random sequence, is at most*

$$2 \cdot \left(1 + \sum_{i=1}^{k-1} 1/\max\{\epsilon^{-1} - i, 1\}\right) \doteq 2\Phi(\epsilon, k).$$

(The "1" is for the m new requests. Note that after the kth non-redundant request, the insurance payment is 0 because the next non-redundant request is the new request starting the next phase.) This gives the deterministic upper bound stated in Theorem 1.1.

3 Lower Bound on Deterministic Strategies

In this section we show that for any deterministic on-line strategy A, there is a distribution D in Δ_ϵ such that the expected cost of the strategy divided by the expected cost of OPT is at least $\Phi(\epsilon, k) - 1$.

3.1 The Adversary

We describe D by describing an adversary that requests pages probabilistically subject to the limitations of Δ_ϵ. Fix $\epsilon > 0$ and $k > 0$. Assume $\epsilon > 1/2k$ (otherwise the sought lower bound $\Phi(\epsilon, k) - 1$ is trivially satisfied — it is at most $\Phi(1/2k, k) - 1$, less than 1). Fix $m = \max\{1, \lceil \epsilon^{-1} \rceil - k\}$.

The adversary requests the pages in an on-line fashion, phase by phase. In the first part of each phase, the adversary makes m new requests by assigning probability only to pages not previously requested.

After the first m requests, there are $k+m$ pages that were requested in the previous phase or in this phase so far. Below we restrict our attention to just these pages.

For each remaining request, the adversary assigns a probability to each page as follows. First priority is given to pages that are not in the cache. We call any such page, or a request to it, *worrisome*. Second priority is given to pages that are in the cache and have already been requested. We call any such page, or a request to it, *redundant*. Lowest priority is given to pages that are in the cache and have not yet been requested. We call any such page, or a request to it, *benign*.

Pages are selected in order of priority and assigned as much probability as possible, subject to the constraint that no page is assigned probability more than ϵ and the total probability assigned is 1. By the choice of m, we have $(k + m)\epsilon \geq 1$, so all three kinds of pages suffice for all probability to be assigned.

The adversary follows this strategy until k distinct pages have been requested, at which point the adversary begins a new phase. This defines the distribution $D \in \Delta_\epsilon$.

3.2 Lower Bound on Expected On-Line Cost

Next we bound from below the expected cost incurred by A within each phase of a random sequence x. We will compare this bound to the expected amortized cost incurred by OPT within the phase to obtain our final lower bound on the competitive ratio.

Consider any phase. Consider those requests that are to pages that have not been requested before during the phase. For each i such that $1 \le i \le k$, let p_i denote the ith such request. Partition the phase into k *runs* according to the p_i's: the first run is the single request p_1, and in general, the ith run starts after the $i-1$st run ends and continues through request p_i.

The first m runs are simply the first m single requests to new pages. Each causes A to fault. We next give a lower bound on the expected cost of each remaining run.

For any ith run, where $m+1 \le i \le k$, let W^i denote the event that the first non-redundant request is worrisome. Then $m + \sum_{i=m+1}^{k} \Pr[W^i]$ is a lower bound on the expected cost of the phase. Fix a particular ith run; we will bound $\Pr[W^i]$.

Let W_j^i, and B_j^i denote the events that the jth request in the run is worrisome or benign, respectively. Then

$$\begin{aligned} \Pr[W^i] &= \sum_j \Pr[W_j^i \cup B_j^i] \Pr[W_j^i \mid W_j^i \cup B_j^i] \\ &\ge \min_j \Pr[W_j^i \mid W_j^i \cup B_j^i]. \end{aligned}$$

This means that it suffices to bound, for each request in the run, the probability that the request is worrisome, conditioned on the request being worrisome or benign. We do this next.

Before a particular jth request in the ith run, let the number of worrisome pages be w. Assume for the moment that the adversary assigns a non-zero probability to some benign page (otherwise the conditional probability is 1). Then the probability that the request is worrisome is ϵw. Further, the probability that the request is benign is at most $1-\epsilon(i-1)$, because each of the $i-1$ pages that has been previously requested in the phase is either worrisome or redundant. Thus, the probability that the request will be worrisome, conditioned on it being worrisome or benign, is at least

$$\frac{\epsilon w}{1-\epsilon(i-1)+\epsilon w} = \frac{w}{\epsilon^{-1}-i+1+w} \ge \frac{m}{\epsilon^{-1}-i+1+m}$$

(because $w \ge m$). Thus, $\Pr[W^i] \ge m/\max\{\epsilon^{-1}-i+1+m, m\}$. (With the "max", this bound holds even if the adversary assigns no probability to the benign pages.) Thus, the expected cost incurred by A during the entire phase is at least

$$\begin{aligned} & m + \sum_{i=m+1}^{k} \frac{m}{\max\{\epsilon^{-1}-i+1+m, m\}} \\ = \; & m + \sum_{i=0}^{k-m-1} \frac{m}{\max\{\epsilon^{-1}-i, m\}} \\ \ge \; & \sum_{i=1}^{k-1} \frac{m}{\max\{\epsilon^{-1}-i, m\}} \\ = \; & \sum_{i=1}^{k-1} \frac{m}{\max\{\epsilon^{-1}-i, 1\}}. \end{aligned}$$

(The last step follows because the choice of m guarantees that either $m = 1$ or $\epsilon^{-1} - i$ is at least m for $i \le k-1$.)

3.3 Comparison to Optimal Cost

Since the amortized cost for OPT during the phase is at most m (see the discussion in § 2.1), we have:

LEMMA 3.1. *On a random sequence from the adversary $D \in \Delta_\epsilon$ described above the expected cost for any deterministic online algorithm A, divided by the expected cost for* OPT, *is at least*

$$\sum_{i=1}^{k-1} 1/\max\{\epsilon^{-1}-i, 1\} \doteq \Phi(\epsilon, k) - 1.$$

This gives the lower bound stated in Theorem 1.1.

4 Randomized Strategies

In this section we extend the lower bound above to randomized strategies. Very little work is required to get lower bounds that match the best possible upper bounds within a factor of roughly two.

Fix $\epsilon > 0$ and $k > 0$. For simplicity we make the following *technical assumption*: ϵ^{-1} is an integer and $\epsilon \le 1/(k+1)$.

Consider the adversary described in the previous section for deterministic strategies. Within a phase, that adversary makes requests to m new pages, followed by requests restricted to a set of $k+m$ pages, where $m = \max\{1, \epsilon^{-1} - k\}$. We make the following key observation about this adversary:

> *Under the technical assumption, the adversary's actions are independent of the deterministic on-line algorithm A.*

The reason: the technical assumption implies that $\epsilon(k+m)$ equals 1. In this case, for each non-new request, the adversary assigns probability ϵ uniformly to each of the $k+m$ pages. In other words, each phase consists of requests to m new pages, followed by a sequence of requests to the $k+m$ pages, where each request is chosen uniformly at random from those pages, until a total of k distinct pages have been requested, after which the next phase begins.

This case turns out to be a generalization of a lower bound on the competitive ratio of randomized on-line strategies against the standard adversary [4, Thm. 13.2]. That lower bound (equivalent to our case $m = 1$) uses the following principle: against any input distribution D, any randomized algorithm R satisfies

$$\mathrm{E}_D[R(x)] \geq \min_A \mathrm{E}_D[A(x)].$$

Here x is a random input from D, and A ranges over all deterministic strategies. Recall that $R(x)$ and $A(x)$ denote the cost of algorithms R and A respectively on input x. This principle follows from the fact that R may be viewed as a probability distribution over the class of deterministic algorithms.

This is the point of the key observation made above: under our technical assumption, the distribution D defined in the previous section turns out to be independent of the on-line algorithm A. Thus, under the technical assumption, the lower bound extends to randomized algorithms:

LEMMA 4.1. *Fix any $\epsilon > 0$ such that ϵ^{-1} is an integer greater than k.*

Then the lower bound on the competitive ratio established for deterministic strategies in Lemma 3.1 also applies to randomized strategies. Namely, on a random input from the adversary $D \in \Delta_\epsilon$ described in the previous section, the expected cost for any randomized online algorithm, divided by the expected cost for OPT, *is at least*

$$\sum_{i=1}^{k-1} 1/\max\{\epsilon^{-1} - i, 1\} \doteq \Phi(\epsilon, k) - 1.$$

This gives the first lower bound stated in Theorem 1.2. The second lower bound follows from the fact that decreasing ϵ only weakens the adversary, so that for $\epsilon \leq 1/(k+1)$, the ratio is at most $\Phi(1/(k+1), k) - 1$, which equals $H(k) - 1$.

The upper bounds follow from the known upper bound of $2H(k)$ on the competitive ratio of MARK against the standard adversary [2, 6, 7], and from the fact that Lemma 2.2 also applies to MARK (since the upper bound applies to any deterministic marking algorithm, i.e., any conditioning of MARK on a particular outcome of its random choices).

References

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

[2] Amos Fiat, Richard M. Karp, Michael Luby, Lyle A. McGeoch, Daniel D. Sleator, and Neal E. Young. Competitive paging algorithms. *Journal of Algorithms*, 12(4):685–699, December 1991.

[3] Elias Koutsoupias and Christos H. Papadimitriou. Beyond competitive analysis. In *Proc. of the 35th IEEE Annual Symp. on Foundation of Computer Science*, pages 394–400, 1994.

[4] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

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

[6] Neal Young. Competitive paging and dual-guided algorithms for weighted caching and matching. (Thesis) Tech. Rep. CS-TR-348-91, Computer Science Department, Princeton University, October 1991.

[7] Neal Young. On-line caching as cache size varies. In *Proc. of the Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 241–250, 1991.