Paging with succinct predictions

Antoniados Antoniadis[1] Joan Boyar[2] Marek Eliáš[3] Lene M. Favrholdt[2]
Ruben Hoeksma[1] Kim S. Larsen[3] Adam Polak[4] Bertrand Simon[5]

[1] University of Twente, Enschede, Netherlands
[2] University of Southern Denmark, Odense, Denmark
[3] Bocconi University, Milan, Italy
[4] EPFL, Lausanne, Switzerland
[5] IN2P3 Computing Center and CNRS, Villeurbanne, France

Keywords : *paging, online algorithms, learning-augmented algorithms.*

The paging (also called *caching*) problem, is a fundamental online problem [3] modeling the management of an internal small but fast memory space in operating systems. A *cache* composed of k identical slots is available, where each slot allows to store one *page*. An instance is described by the size of the cache k, a universe of pages, and an online page sequence, describing requested pages which must be served when they arrive. These page requests arrive one at a time, and the algorithm must ensure that this page is stored in the cache. If the page is absent, a *page miss* occurs and the algorithm must pay a unit cost, load this page into cache, and potentially evict a previously loaded page if the cache was full. If the page was already present in the cache, a *page hit* occurs and the request is served with no cost. The objective is then to derive an algorithm deciding which page to evict when page misses happen, aiming at minimizing the total number of page misses. Of course, future page requests are unknown to the algorithm as this is an online problem.

The classic metric to estimate the quality of an online algorithm consists in computing its *competitive ratio*. An algorithm A has a competitive ratio α if and only if there exists a constant c such that for any instance I, $\textsc{Alg}(I) \leq \alpha \cdot \textsc{Opt}(I) + c$, where $\textsc{Alg}(I)$ and $\textsc{Opt}(I)$ represent respectively the cost of A and of an optimal offline algorithm on I (which knows the whole instance beforehand).

The paging problem is now well-understood under the scope of competitive analysis. An optimal offline solution consists in following the so-called Belady's rule: always evict the page requested the furthest in the future. Deterministic solutions are k-competitive at best, such as FIFO (First In, First Out: evicting the page which entered the cache the furthest in the past), LRU (Least Recently Used: evicting the page requested the furthest in the past), or a special class of algorithms called *marking*. Randomized algorithm cannot be better than H_k-competitive, where $H_k \approx \ln k$ is the k-th harmonic number. The well-known *marker* algorithm is $(2H_k - 1)$ competitive [4] and algorithms matching the lower bound have been discovered later [1].

However, it is notorious that competitive analysis leads to pessimistic algorithms that try to prepare for pathological cases and fail to obtain a good solution on easy instances. This is one of the motivations of a quite recent field called *learning-augmented algorithms*. We assume here that algorithms get access to *predictions* about the future of the instance, which typically come from machine learning models able to recognize patterns in the input. These prediction may be inaccurate, and the main idea is to design algorithm which can be run regardless of the quality of the predictions. Algorithms are generally evaluated using three quantities: *robustness* corresponds to the classic competitive ratio when the predictions are irrelevant; *consistency* is the competitive ratio assuming all predictions are perfect; and *smoothness* corresponds to how fast the competitive ratio increases when the predictions errors increase. A large amount of problems have been studied under this scope within a few years, as hundreds of papers have been attributed to it [5].

The paging problem is one of the problems which has been the most studied under this scope, and several predictions models have been considered. The seminal paper from Lykouris and Vassilvitskii [6] assume that at each round, a prediction about the next time the current page will be requested again is available, their results being improved later. Other models include predictions about all requests until the next reoccurrence, about the relative order of page requests, or about the cache content of an optimal algorithm.

The objective of this paper (full version available at [2]) is to study the usage of *succinct* predictions for the paging problem, so that are composed of few bits of information. Apart from the theoretical interest, the motivation comes from the faster communications and computations involved, and existing binary classifiers outputting a single bit of information per page. We propose two models in which a 1-bit-prediction arrives with each page request, which is, up to constant factors, necessary to improve worst-case guarantees. In the *discard* model, the predictions describe whether a given optimal algorithm would evict the page before it is requested again. In the *phase* model, the predictions describe whether this page is requested in the next k-phase (defined as in marking algorithms).

Denoting by η_0 (resp. η_1) the number of pages wrongly predicted 0 (resp. 1), Table 1 summarizes the guarantees achieved by our algorithms, which are shown to be close to the best possible by corresponding lower bounds. The asymmetry of η_0 and η_1 is remarkable: it is much more expensive to wrongly keep a page in cache rather than to wrongly evict it.

	Discard predictions	Phase predictions
Deterministic	$\textsc{Opt} + (k-1)\eta_0 + \eta_1$	*not suited for deterministic*
Randomized	$\textsc{Opt} + 2H_k \cdot \eta_0 + \eta_1$	$2\textsc{Opt} + H_k \cdot \eta_0 + \eta_1$ $2\left(\ln\left(\frac{2\eta_1}{\textsc{Opt}} + 1\right) + 2\right) \cdot \textsc{Opt} + H_k \cdot \eta_0$

TAB. 1: Summary of results: upper bounds on the cost of our algorithms.

References

[1] D. Achlioptas, M. Chrobak, and J. Noga. Competitive analysis of randomized paging algorithms. *Theoretical Computer Science*, 234(1–2):203–218, 2000.

[2] Antonios Antoniadis, Joan Boyar, Marek Eliáš, Lene Monrad Favrholdt, Ruben Hoeksma, Kim S Larsen, Adam Polak, and Bertrand Simon. Paging with succinct predictions. In *International Conference on Machine Learning*, pages 952–968. PMLR, 2023.

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

[4] A. Fiat, R. M. Karp, M. Luby, L. A. McGeoch, D. D. Sleator, and N. E. Young. Competitive paging algorithms. *Journal of Algorithms*, 12:685–699, 1991.

[5] Alexander Lindermayr and Nicole Megow. Algorithms with predictions. `https://algorithms-with-predictions.github.io`, 2022. [Online; accessed 8-September-2022].

[6] Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. *J. ACM*, 68(4):24:1–24:25, 2021.