Online Paging with a Vanishing Regret

Yuval Emek
Faculty of Industrial Engineering and Management,
Technion – Israel Institute of Technology, Haifa, Israel
yemek@technion.ac.il

Shay Kutten
Faculty of Industrial Engineering and Management,
Technion – Israel Institute of Technology, Haifa, Israel
kutten@technion.ac.il

Yangguang Shi
Faculty of Industrial Engineering and Management,
Technion – Israel Institute of Technology, Haifa, Israel
shiyangguang@campus.technion.ac.il

Abstract

This paper considers a variant of the online *paging* problem, where the online algorithm has access to multiple *predictors*, each producing a sequence of predictions for the page arrival times. The predictors may have occasional prediction errors and it is assumed that at least one of them makes a sublinear number of prediction errors in total. Our main result states that this assumption suffices for the design of a randomized online algorithm whose time-average *regret* with respect to the optimal offline algorithm tends to zero as the time tends to infinity. This holds (with different regret bounds) for both the *full information* access model, where in each round, the online algorithm gets the predictions of all predictors, and the *bandit* access model, where in each round, the online algorithm queries a single predictor.

While online algorithms that exploit inaccurate predictions have been a topic of growing interest in the last few years, to the best of our knowledge, this is the first paper that studies this topic in the context of multiple predictors for an online problem with unbounded request sequences. Moreover, to the best of our knowledge, this is also the first paper that aims for (and achieves) online algorithms with a vanishing regret for a classic online problem under reasonable assumptions.

2012 ACM Subject Classification Theory of computation → Online learning algorithms; Theory of computation → Caching and paging algorithms

Keywords and phrases online paging, inaccurate predictions, multiple predictors, vanishing regret, full information vs. bandit access

Digital Object Identifier 10.4230/LIPIcs.ITCS.2021.67

Related Version A full version of the paper is available at `https://arxiv.org/abs/2011.09439`.

Funding *Yuval Emek*: The work of Yuval Emek was supported in part by an Israeli Science Foundation grant number 1016/17.
Shay Kutten: The work of Shay Kutten was supported in part by a grant from the ministry of science in the program that is joint with JSPS and in part by the BSF.
Yangguang Shi: The work of Yangguang Shi was partially supported at the Technion by a fellowship of the Israel Council for Higher Education.

1 Introduction

A critical bottleneck in the performance of digital computers, known as the “memory wall”, is that the main memory (a.k.a. DRAM) is several orders of magnitude slower than the multiprocessor [26, 16, 4]. Modern computer architectures bridge this performance gap by utilizing a cache, namely, a memory structure positioned next to the multiprocessor that responds much faster than the main memory. However, the cache is inherently smaller than the main memory which means that some of the memory items requested by the running program may be missing from the cache. When such a *cache miss* occurs, the multiprocessor is required to fetch the requested item from the main memory into the cache; if the cache is already full, then some previously stored item must be evicted to make room for the new one. Minimizing the number of cache misses is known to be a primary criterion for improving the computer's performance [26, 17].

The aforementioned challenge is formalized by means of a classic *online* problem called *paging* [21] (a.k.a. unweighted *caching*), defined over a main memory that consists of $n \in \mathbb{Z}_{>0}$ *pages* and a cache that holds $k \in \mathbb{Z}_{>0}$ pages at any given time, $k < n$. The execution of a paging algorithm $\texttt{Alg}$ progresses in $T \in \mathbb{Z}_{>0}$ discrete *rounds*, where round $t \in T$ occupies the time interval $[t, t-1)$. An instance of the paging problem is given by a sequence $\sigma = \{\sigma_t\}_{t\in[T]}$ of page *requests* so that request $\sigma_t \in [n]$ is revealed at time $t \in [T]$. Denoting the *cache configuration* of $\texttt{Alg}$ at time t by $C_t \subset [n]$, $|C_t| = k$, if $\sigma_t \in C_t$, then $\texttt{Alg}$ does nothing in round t; otherwise ($\sigma_t \notin C_t$), a *cache miss* occurs and $\texttt{Alg}$ should bring the requested page into the cache so that $\sigma_t \in C_{t+1}$. Since $|C_{t+1}| = |C_t| = k$, it follows that upon a cache miss, $\texttt{Alg}$ must evict some page $i \in C_t$ and its policy is reduced to the selection of this page i. The *cost* incurred by $\texttt{Alg}$ on σ is defined to be the number of cache misses it suffers throughout the execution, denoted by

$$\text{cost}_\sigma(\texttt{Alg}) \;=\; |\{t \in [T] : \sigma_t \notin C_t\}| \;,$$

taking the expectation if $\texttt{Alg}$ is a randomized algorithm. When σ is clear from the context, we may omit the subscript, writing $\text{cost}(\texttt{Alg}) = \text{cost}_\sigma(\texttt{Alg})$.

NAT and the FitF Algorithm. To avoid cumbersome notation, we assume hereafter that the request sequence σ is augmented with a suffix of n virtual requests so that $\sigma_{T+i} = i$ for every $i \in [n]$. This facilitates the definition of the *next arrival time (NAT)* of page $i \in [n]$ with respect to time $t \in [T]$ as the first time after t at which page i is requested, denoted by

$$A_t(i) \;=\; \min\{t' > t \mid \sigma_{t'} = i\} \,.$$

Based on that, we can define the $\texttt{FitF}$ (stands for furthest in the future) paging algorithm that on a cache miss at time $t \in [T]$, evicts the page $i \in C_t$ that maximizes $A_t(i)$. A classic result of Belady [5] states that $\texttt{FitF}$ is optimal in terms of the cost it incurs for the given request sequence σ; we subsequently denote $\texttt{OPT}_\sigma = \text{cost}_\sigma(\texttt{FitF})$ and omit the subscript, writing $\texttt{OPT} = \texttt{OPT}_\sigma$, when σ is clear from the context. It is important to point out that $\texttt{FitF}$ is an *offline* algorithm as online algorithms are oblivious to the NATs.

Regret. We define the *regret* of an online paging algorithm $\texttt{Alg}$ on σ as

$$\text{regret}_\sigma(\texttt{Alg}) \;=\; \text{cost}_\sigma(\texttt{Alg}) - \texttt{OPT}_\sigma$$

and omit the subscript, writing $\text{regret}(\texttt{Alg}) = \text{regret}_\sigma(\texttt{Alg})$ when σ is clear from the context. Our goal in this paper is to develop an online algorithm that admits a *vanishing regret*, namely, an online algorithm $\texttt{Alg}$ for which it is guaranteed that

$$\lim_{T\to\infty} \frac{\sup\left\{\text{regret}_\sigma(\texttt{Alg}) \mid \sigma \in [n]^T\right\}}{T} \;=\; 0 \,.$$

The following theorem states that this goal is hopeless unless the online algorithm has access to some additional information; its proof should be a folklore, we add it in the full version [11] of this paper for completeness.

▶ **Theorem 1.** *Fix $n = k+1$ and let σ be a request sequence generated by picking σ_t uniformly at random (and independently) from $[n]$ for $t = 1, \ldots, T$. Then, $\mathbb{E}(\text{cost}(\texttt{Alg})) \geq \Omega\left(\frac{T}{k}\right)$ for any (possibly randomized) online paging algorithm* `Alg`*, whereas $\mathbb{E}(\texttt{OPT}) \leq O\left(\frac{T}{k \log k}\right)$.*

1.1 Machine Learned Predictions

Developments in *machine learning (ML)* technology suggest a new direction for reducing the number of cache misses by means of predicting the request sequence. Indeed, recent studies have shown that neural networks can be employed to predict the memory pages accessed by a program with high accuracy [16, 9, 22, 19, 23]. When provided with an accurate prediction of the request sequence σ, one can simply simulate `FitF`, thus ensuring an optimal performance.

Unfortunately, the predictions generated by ML techniques are usually not 100% accurate as a result of a distribution drift between the training and test examples or due to adversarial examples [24, 18]. This gives rise to a growing interest in developing algorithmic techniques that can overcome inaccurate predictions, aiming for the design of online algorithms with performance guarantee that improves as the predictions become more accurate [18, 20, 1, 25]. The existing literature in this line of research studies a setting where the online algorithm `Alg` is provided with a sequence of predictions for σ and focuses on bounding `Alg`'s *competitive ratio* as a function of the proximity of this sequence to σ (more on that in Section 2).

The current paper tackles the challenge of overcoming inaccurate predictions from a different angle: Motivated by the abundance of forecasting algorithms that may be trained on different data sets or using different models (e.g., models that are robust to adversarial examples [15]), we consider a decision maker with access to *multiple* predicting sequences for σ. Our main goal is to design an online algorithm `Alg` that admits a vanishing regret assuming that at least one of the predicting sequences is sufficiently accurate, even though the decision maker does not know in advance which predicting sequence it is.

Explicit Predictors. Formally, we consider $M \in \mathbb{Z}_{>0}$ *predictors* whose role is to predict the request sequence σ. In the most basic form, referred to hereafter as the *explicit predictors* setting, each predictor $j \in [M]$ produces a page sequence $\pi^j = \{\pi_t^j\}_{t \in [T]} \in [n]^T$, where π_t^j aims to predict σ_t for every $t \in [T]$, and the sequences $\pi^1, \ldots, \pi^M$ are revealed to the online algorithm `Alg` at the beginning of the execution. Under the explicit predictors setting, predictor $j \in [M]$ is said to have a *prediction error* in round $t \in [T]$ if $\pi_t^j \neq \sigma_t$. We measure the accuracy of predictor j by means of her *cumulative prediction error*

$$\eta_e^j = \eta_e(\pi^j) = \left|\left\{t \in [T] : \pi_t^j \neq \sigma_t\right\}\right|$$

and define $\eta_e^{\min} = \min\{\eta_e^j \mid j \in [M]\}$.

The fundamental assumption that guides the current paper, referred to hereafter as the *good predictor* assumption, is that there exists at least one predictor whose cumulative prediction error is sublinear in T, namely, $\eta_e^{\min} = o(T)$. We emphasize that `Alg` has no a priori knowledge of $\eta_e^1, \ldots, \eta_e^M$ nor does it know the predictor that realizes $\eta_e^{\min}$. Our main research question can now be stated as follows:

> Does the good predictor assumption provide a sufficient condition for the existence of an online algorithm that admits a vanishing regret?

NAT Predictors. For the paging problem, it is arguably more natural to consider the setting of *NAT predictors*, where predictor $j \in [M]$ produces in each round $t \in [T]$, a prediction $a_t^j \in (t, T+n]$ for the NAT $A_t(\sigma_t)$ of the page that has just been requested. Under this setting, predictor $j \in [M]$ is said to have a *prediction error* in round $t \in [T]$ if $a_t^j \neq A_t(\sigma_t)$. As in the explicit predictors setting, we measure the accuracy of (NAT) predictor j by means of her *cumulative prediction error*, now defined as

$$\eta_N^j = \left| \left\{ t \in [T] : a_t^j \neq A_t(\sigma_t) \right\} \right| \tag{1}$$

(this measure is termed *classification loss* in [18]), and define $\eta_N^{\min} = \min\{\eta_N^j \mid j \in [M]\}$.[1] The NAT predictors version of the good predictor assumption states that $\eta_N^{\min} = o(T)$.

Given a page sequence $\pi = \{\pi_t\}_{t \in [T]} \in [n]^T$ augmented with a suffix of n pages such that $\pi_{T+i} = i$ for every $i \in [n]$, we say that (NAT) predictor $j \in [M]$ is *consistent* with π if $a_t^j = \min\{t' > t \mid \pi_{t'} = \sigma_t\}$ for every $t \in [T]$; if the page sequence π is not important or clear from the context, then we may say that predictor j is *consistent* without mentioning π. The key observation here is that if predictor j is consistent with a page sequence π, then η_N^j provides a good approximation for $\eta_e(\pi)$, specifically,

$$\eta_e(\pi) - n \leq \eta_N^j \leq 2 \cdot \eta_e(\pi) \tag{2}$$

(the proof is deferred to the full version [11]). This means that the setting of NAT predictors is stronger than that of explicit predictors in the sense that NAT predictor $j \in [M]$ can be simulated (consistently) from explicit predictor j by deriving the NAT prediction a_t^j in round $t \in [T]$ from the (explicit) predictions $\pi_{t+1}^j, \pi_{t+2}^j, \ldots, \pi_T^j$, while ensuring that η_N^j is a good approximation for η_e^j. Therefore, unless stated otherwise, we subsequently restrict our attention to NAT predictors and in particular omit the subscript from the cumulative prediction error notation, writing $\eta^j = \eta_N^j$ and $\eta^{\min} = \eta_N^{\min}$. It is important to point out though that the results established in the current paper hold regardless of whether the (NAT) predictors are consistent or not.

Access Models. Recall that the (NAT) predictors $j \in [M]$ produce their predictions in an online fashion so that the NAT prediction a_t^j is produced in round t. This calls for a distinction between two *access models* that determine the exact manner in which a_t^j is revealed to the online paging algorithm `Alg`. First, we consider the *full information* access model, where in each round $t \in [T]$, `Alg` receives a_t^j for all $j \in [M]$. Motivated by systems in which accessing the ML predictions is costly in both time and space (thus preventing `Alg` from querying multiple predictors in the same round and/or predictions belonging to past rounds), we also consider the *bandit* access model, where in each round $t \in [T]$, `Alg` receives a_t^j for a single predictor $j \in [M]$ selected by `Alg` in that round. To make things precise, we assume, under both access models, that if `Alg` has to evict a page in round t, then the decision on the evicted page is made prior to receiving the prediction(s) in that round. Notice though that the information that `Alg` receives from the predictor(s) is not related to the evicted page and as such, should not be viewed as a feedback that `Alg` receives in response to the action it takes in the current round.

1.2 Our Contribution

Consider a (single) predictor that in each round $t \in [T]$, produces a prediction a_t for the NAT $A_t(\sigma_t)$ of the page that has just been requested and let η be her cumulative prediction error. Our first technical contribution comes in the form pf a thorough analysis of the performance

[1] In Section 2, we provide a refined definition for the cumulative prediction error of a NAT predictor that is more robust against adversarial interference such as shifting each a_j^t by a constant. For simplicity of the exposition, the definition presented in Eq. (1) is used throughout the current section; we emphasize though that all our results hold for the stronger notion of prediction error as defined in Section 2.

of a simple online paging algorithm called `Sim` that simulates `FitF`, replacing the actual NATs with the ones derived from the prediction sequence $\{a_t\}_{t \in [T]}$. Using some careful combinatorial arguments, we establish the following bound.

▶ **Theorem 2.** *The regret of* `Sim` *satisfies* $\text{regret}(\texttt{Sim}) \leq O(\eta + k)$.

Relying on online learning techniques, Blum and Burch [6] develop an online algorithm "multiplexer" that given multiple online algorithms as subroutines, produces a randomized online algorithm that performs almost as good as the best subroutine in hindsight. Applying Theorem 2 to the M predictors so that each predictor $j \in [M]$ yields its own online paging algorithm $\texttt{Sim}^j$ and plugging algorithms $\texttt{Sim}^1, \dots, \texttt{Sim}^M$ into the multiplexer of [6], we establish the following theorem, thus concluding that the good predictor assumption implies an online paging algorithm with a vanishing regret under the full information access model.

▶ **Theorem 3.** *There exists a randomized online paging algorithm that given full information access to M NAT predictors with minimum cumulative prediction error $\eta^{\min}$, has regret at most $O\left(\eta^{\min} + k + (Tk \log M)^{1/2}\right)$.*

Combined with (2), we obtain the same asymptotic regret bound for explicit predictors.

▶ **Corollary 4.** *There exists a randomized online paging algorithm that given access to M explicit predictors with minimum cumulative prediction error $\eta_e^{\min}$, has regret at most $O\left(\eta_e^{\min} + k + (Tk \log M)^{1/2}\right)$.*

The explicit predictors setting is general enough to make it applicable to virtually any online problem. This raises the question of whether other online problems admit online algorithms with a vanishing regret given access to explicit predictors whose minimum cumulative prediction error is sublinear in T. We view the investigation of this question as an interesting research thread that will hopefully arise from the current paper.

Going back to the setting of NAT predictors, one wonders if a vanishing regret can be achieved also under the bandit access model since the technique of [6] unfortunately does not apply to this more restricted access model. An inherent difficulty in the bandit access model is that we cannot keep track of the cache configuration of $\texttt{Sim}^j$ unless predictor j is queried in each round (which means that no other predictor can be queried). To overcome this obstacle, we exploit certain combinatorial properties of the `Sim` algorithm to show that $\texttt{Sim}^j$ can be "chased" without knowing its current cache configuration, while bounding the accumulated cost difference. By a careful application of online learning techniques, this allows us to establish the following theorem, thus concluding that the good predictor assumption implies an online paging algorithm with a vanishing regret under the bandit access model as well.

▶ **Theorem 5.** *There exists a randomized online paging algorithm that given bandit access to M NAT predictors with minimum cumulative prediction error $\eta^{\min}$, has regret at most $O\left(\eta^{\min} + T^{2/3} k M^{1/2}\right)$.*

1.3 Related Work and Discussion

We say that an online algorithm `Alg` for a minimization problem $\mathcal{P}$ has *competitive ratio* α if for any instance σ of $\mathcal{P}$, the cost incurred by `Alg` on σ is at most $\alpha \cdot \texttt{OPT}_\sigma + \beta$, where $\texttt{OPT}_\sigma$ is the cost incurred by an optimal offline algorithm on σ and β is a constant that may depend on $\mathcal{P}$, but not on σ [21, 7]. In comparison, the notion of regret as defined in the current paper uses the optimal offline algorithm as an absolute (additive), rather than relative (multiplicative), benchmark. Notice that the vanishing regret condition cannot be expressed in the scope of the competitive ratio definition. In particular, $\alpha = 1$ is a stronger requirement than vanishing regret as the latter can accommodate an additive parameter β that does depend on σ as long as it is sublinear in $T = |\sigma|$. On the other hand, $\alpha > 1$ implies a non-vanishing regret when $\mathtt{OPT}_\sigma$ scales linearly with T.

As mentioned in Section 1.1, most of the existing literature on augmenting online algorithms with ML predictions is restricted to the case of a single predictor [18, 20, 1, 25]. The goal of these papers is to develop online algorithms with two guarantees: (i) their competitive ratio tends to $O(1)$ (though not necessarily to 1) as the predictor's accuracy improves; and (ii) they are robust in the sense that regardless of the predictor's accuracy, their competitive ratio is not much worse than that of the best online algorithm that has no access to predictions.

In contrast, the current paper addresses the setting of multiple predictors, working under the assumption that at least one of them is sufficiently accurate, and seeking to develop online algorithms with a vanishing regret. To the best of our knowledge, this is the first paper that aims at this direction.

Most closely related to the current paper are the papers of [18, 20, 25] on online paging with predictions. The authors of these papers stick to the setting of a (single) NAT predictor and quantify the predictor's accuracy by means of the L_1-*norm*. Specifically, taking $\{a_t\}_{t\in[T]}$ to be the sequence of NAT predictions, they define the predictor's cumulative prediction error to be $\sharp\mathtt{L}_1 = \sum_t |a_t - A_t(\sigma_t)|$. It is easy to see that for any NAT predictor, the cumulative prediction error as defined in (1) is never larger than its $\sharp\mathtt{L}_1$, while the former can be $\Omega(T)$-times smaller.

In particular, Lykouris and Vassilvitskii [18] design a randomized online paging algorithm whose competitive ratio is at most $O\left(\min\left\{1 + \sqrt{\sharp\mathtt{L}_1/\mathtt{OPT}}, \log k\right\}\right)$. Rohatgi [20] presents an improved randomized online algorithm with competitive ratio up-bounded by $O\left(\min\left\{1 + \frac{\log k}{k}\frac{\sharp\mathtt{L}_1}{\mathtt{OPT}}, \log k\right\}\right)$ and accompany this with a lower bound of $\Omega\left(\min\left\{1 + \frac{1}{k\log k}\frac{\sharp\mathtt{L}_1}{\mathtt{OPT}}, \log k\right\}\right)$. Notice that the online algorithms presented in [18, 20] belong to the *marking family* of paging algorithms [13] and it can be shown that the competitive ratio of any such algorithm is bounded away from 1 even when provided with a fully accurate predictor (consider for example the paging instance defined by setting $n = 4$, $k = 2$, and $\sigma_t = (t \bmod 4) + 1$ for every $t \in [T]$).

Recently, Wei [25] advanced the state of the art of this problem further, presenting a randomized $O\left(\min\left\{1 + \frac{1}{k}\frac{\sharp\mathtt{L}_1}{\mathtt{OPT}}, \log k\right\}\right)$-competitive online paging algorithm. To do so, Wei analyzes an algorithm called *BlindOracle*, that can be viewed as a variant of our `Sim` algorithm (see Section 1.2), and proves that its competitive ratio is at most $\min\left\{1 + O\left(\frac{\sharp\mathtt{L}_1}{\mathtt{OPT}}\right), O\left(1 + \frac{1}{k}\frac{\sharp\mathtt{L}_1}{\mathtt{OPT}}\right)\right\}$. He then plugs this algorithm into the multiplexer of [6] together with an $O(\log k)$-competitive off-the-shelf randomized online paging algorithm to obtain the promised competitive ratio. Notice that the bound that Wei establishes on the competitive ratio of BlindOracle immediately implies an $O(\sharp\mathtt{L}_1)$ bound on the regret of this algorithm. As such, Theorem 2 can be viewed as a refinement of Wei's result, bounding the regret as a function of η rather than the weaker measure of $\sharp\mathtt{L}_1$.

Antoniadis et al. [1] studies online algorithms with ML predictions in the context of the *metrical task system (MTS)* problem [8]. They consider a different type of predictor that in each round $t \in [T]$, provides a prediction $\hat{s}_t$ for the state s_t of an optimal offline algorithm, measuring the prediction error by means of $\sharp\mathtt{Distances} = \sum_{t\in[T]} \operatorname{dist}(s_t, \hat{s}_t)$, where $\operatorname{dist}(\cdot,\cdot)$ is the distance function of the underlying metric space. It is well known that any paging instance $\mathcal{I}$ can be transformed into an MTS instance $\mathcal{I}_{\text{MTS}}$. Antoniadis et al. prove that the prediction sequence $\{a_t\}_{t\in[T]}$ of a NAT predictor for $\mathcal{I}$ can also be transformed into a prediction sequence $\{\hat{s}_t\}_{t\in[T]}$ for $\mathcal{I}_{\text{MTS}}$. However, the resulting prediction error $\sharp\texttt{Distances}$ of the latter sequence is incomparable to the prediction error $\sharp\texttt{L}_1$ of the former; this remains true also for the stronger notion of cumulative prediction error as defined in (1).

Online algorithms with access to multiple predictors have been studied by Gollapudi and Panigrahi for the *ski rental* problem [14]. Among other results, they prove that a competitive ratio of $\alpha = \frac{4}{3}$ (resp., $\alpha = \frac{\sqrt{5}+1}{2}$) can be achieved by a randomized (resp., deterministic) online algorithm that has access to two predictors assuming that at least one of them provides an accurate prediction for the number of skiing days. Notice that the length of the request sequence in the ski rental problem is inherently bounded by the cost of buying the ski gear; this is in contrast to the paging problem considered in the current paper, where much of the challenge comes from the unbounded request sequence.

The reader may have noticed that some of the terminology used in the current paper is borrowed from the *online learning* domain [10]. The main reason for this choice is that the research objectives of the current paper are, to a large extent, more in line with the objectives common to the online learning literature than they are in line with the objectives of the literature on online computation. In particular, as discussed already, we measure the quality of our online algorithms by means of their regret (rather than competitiveness), indicating that the online algorithm can be viewed as a decision maker that tries to learn the best offline algorithm.

1.4 Paper's Organization

The remainder of this paper is organized as follows. In Section 2, we refine the notion of cumulative prediction error as defined in (1) and compare the refined notion with the number of inversions used in some of the related literature [20, 25]. The analysis of the `Sim` algorithm (using a single predictor), leading to the proof of Theorem 2, is carried out in Section 3. Section 4 is then dedicated to the setting of multiple (NAT) predictors under the bandit access model and establishes Theorem 5. As discussed in Section 1.3, Theorem 3, dealing with the full information access model, follows from Theorem 2 combined with a technique of [6]; this is explained in more detail in the full version [11].

2 Measurements of Prediction Errors

The measurement of the prediction errors plays an important role in the study on online paging algorithms augmented by predictions. This part makes a comparison between different measurements for the scenario where there is a single NAT predictor. To avoid ambiguity, in this part we use $\sharp\texttt{ErrorRounds}$ to represent the measurement defined in Eq. (1) for the single predictor j. In the following, the superscript j for the predictor is omitted for convenience.

In the analysis of [25], the prediction errors are measured with the number of *inverted pairs*. For a pair of two rounds $\{t, t'\}$, we say it is an inverted pair if $A_t(\sigma_t) < A_{t'}(\sigma_{t'})$ and $a_t \geq a_{t'}$. Let $\texttt{INV}$ be the set of all the inverted pairs, and define $\sharp\texttt{InvertedPairs} = |\texttt{INV}|$. To compare the measurement $\sharp\texttt{InvertedPairs}$ with $\sharp\texttt{ErrorRounds}$, we also define the following notations.

$$\sharp\texttt{InvertedRounds} \doteq \Big|\{t \mid \exists t' \text{ s.t. } \{t,t'\} \in \texttt{INV}\}\Big|$$

$$\sharp\texttt{ErrorRoundsInInversion} \doteq \Big|\{t \mid A_t(\sigma_t) \neq a_t \,\wedge\, \exists t' \text{ s.t. } \{t,t'\} \in \texttt{INV}\}\Big|$$

First, it trivially holds that $\sharp\texttt{InvertedRounds} \le 2\cdot\sharp\texttt{InvertedPairs}$, and $\sharp\texttt{InvertedPairs}$ can be $\Omega(T)$ times larger than $\sharp\texttt{InvertedRounds}$. To see the second claim, consider the following sequence σ of requested pages and a prediction sequence π.

$$\sigma_t = \begin{cases} 1 & \text{if } t \le \frac{T}{2} \\ 2 & \text{if } t > \frac{T}{2} \end{cases}, \quad \text{and} \quad \pi_t = \begin{cases} 2 & \text{if } t \le \frac{T}{2} \\ 1 & \text{if } t > \frac{T}{2} \end{cases}.$$

It can be verified that for a sequence of predictions in the form of NATs that are consistent with the settings above, $\sharp\texttt{InvertedPairs}$ is in the order of T^2 while $\sharp\texttt{InvertedRounds}$ is in the order of T.

Second, we claim that $\sharp\texttt{InvertedRounds}$ and $\sharp\texttt{ErrorRounds}$ are incomparable, which means that there exists an example where $\sharp\texttt{InvertedRounds}$ is $\Omega(T)$ times larger than $\sharp\texttt{ErrorRounds}$, and vice versa. Still, we demonstrate these examples with the sequence σ of requests and the prediction sequence π, and the claims above can be verified after converting the predictions in the form of requests to consistent predictions in the form of NATs. The configuration of the first example is given as follows.

$$\sigma_t = \begin{cases} (t \bmod (k-1)) + 1 & \text{if } 1 < t < T \\ k & \text{otherwise} \end{cases}, \quad \text{and} \quad \pi = \begin{cases} (t \bmod (k-1)) + 1 & \text{if } t > 2 \\ k & \text{otherwise} \end{cases}.$$

The second example is configured as follows.

$$\sigma_t = (t \bmod (k-1)) + 1\,, \quad \text{and} \quad \pi = \begin{cases} ((t-1) \bmod (k-1)) + 1 & \text{if } t > 1 \\ k & \text{otherwise} \end{cases}.$$

Third, it is obvious that

$$\sharp\texttt{ErrorRoundsInInversion} \le \min\left\{\sharp\texttt{ErrorRounds}, \sharp\texttt{InvertedRounds}\right\}.$$

Although in Section 1.1 we define η^j for every predictor j in the form of $\sharp\texttt{ErrorRounds}$ for simplicity, our technique indeed works for the better measurement $\sharp\texttt{ErrorRoundsInInversion}$. Therefore, in the technical parts of the current paper, including Section 3 and Section 4, we use the following refined definition of η^j by abuse of notation:

$$\eta^j \doteq \left|\{t \mid A_t(\sigma_t) \ne a_t^j \wedge \exists t' \text{ s.t. } \{t,t'\} \in \texttt{INV}^j\}\right|,$$

where $\texttt{INV}^j = \{(t,t') | A_t(\sigma_t) < A_{t'}(\sigma_{t'}) \wedge a_t^j \ge a_{t'}^j\}$.

3 Single NAT Predictor

We start with the NAT predictor setting with $M = 1$. In such a case, there is no difference between the full information access model and the bandit access model. Throughout this section, we still omit the superscript j for the index of the predictor.

The algorithm $\texttt{Sim}$ that we consider for this setting simulates $\texttt{FitF}$ with maintaining a value $\hat{a}_t(i)$, which we call the *remedy prediction*, for each round $t \in [T]$ and each page $i \in [n]$. In particular, for each page $i \in [n]$, $\texttt{Sim}$ sets

$$\hat{a}_1(i) = \begin{cases} a_1 & \text{if } i = \sigma_1 \\ Z+1 & \text{otherwise} \end{cases}, \quad \text{and}$$

$$\forall\, t \in [2,T] \quad \hat{a}_t(i) = \begin{cases} a_t & \text{if } i = \sigma_t \\ Z & \text{if } \hat{a}_{t-1}(i) \le t \wedge i \ne \sigma_t \wedge \hat{a}_{t-1}(i) \le \hat{a}_{t-1}(\sigma_t) < Z \\ \hat{a}_{t-1}(i) & \text{otherwise} \end{cases}, \tag{3}$$

where $Z > T + n$ is a sufficiently large integer. For each round t when a cache miss happens, the algorithm evicts the page $\hat{e}_t = i$ that maximizes $\hat{a}_t(i)$, and ties are broken in an arbitrary way. The following statements can be directly inferred from Eq. (3).

▶ **Lemma 6.** *The following properties are satisfied for every round $t \in [T]$:*

- *If $a_t > A_t(\sigma_t)$, then for each round $t' \in [t, \min\{T, A_t(\sigma_t) - 1\}]$, we have $\hat{a}_{t'}(\sigma_t) > A_{t'}(\sigma_t)$.*
- *If $a_t < A_t(\sigma_t)$, then for each round $t' \in [t, \min\{T, A_t(\sigma_t) - 1\}]$, either $\hat{a}_{t'}(\sigma_t) < A_{t'}(\sigma_t)$ or $\hat{a}_{t'}(\sigma_t) = Z > A_{t'}(\sigma_t)$. Particularly, if $t' \le a_t - 1$, then $\hat{a}_{t'}(\sigma_t) < A_{t'}(\sigma_t)$.*
- *If $a_t = A_t(\sigma_t)$, then for each round $t' \in [t, \min\{T, A_t(\sigma_t) - 1\}]$, we have $\hat{a}_{t'}(\sigma_t) = A_{t'}(\sigma_t)$.*

Next, we will analyze the cost incurred by `Sim` and show that it has a vanishing regret.

3.1 Definitions and Notations for Analysis

For each round t, we use e_t and $\hat{e}_t$ to represent the pages that are evicted by `FitF` and `Sim`, respectively. We say $e_t = \perp$ (resp. $\hat{e}_t = \perp$) if `FitF` (resp. `Sim`) does not evict any page.

For each page $i \in [n]$ and each round $t \in [T]$, define $R_t(i)$ to be the last round before t when i is requested. Formally,

$$R_t(i) \doteq \begin{cases} \max\{t' < t \mid \sigma_{t'} = i\} & \text{if } \exists\, t' \in [1, t) \text{ s.t. } \sigma_{t'} = i \\ -1 & \text{otherwise} \end{cases} . \tag{4}$$

The following results can be inferred from Eq. (4) and Lemma 6.

For any round $t \in [T]$, let C_t and $\hat{C}_t$ be the *cache profiles* incurred by `FitF` and `Sim`, respectively. More specifically, C_1 and $\hat{C}_1$ represent the cache items given at the beginning. To provide tools for the more complicated scenario where there are multiple predictors, the analysis in this section is carried out without assuming that $C_1 = \hat{C}_1$. For each $t \in [T-1]$, the cache profile C_t (resp. $\hat{C}_t$) is updated to C_{t+1} (resp. $\hat{C}_{t+1}$) immediately after `FitF` (resp. `Sim`) has processed the request σ_t. The cache profiles of `FitF` and `Sim` after serving σ_T are denoted by C_{T+1} and $\hat{C}_{T+1}$, respectively.

Denote the intersection between the cache profiles at each round $t \in [T]$ by $I_t = C_t \cap \hat{C}_t$. Define the *distance* between the cache profiles to be $d_t = k - |I_t|$. We use δ_t to represent the difference in the costs between `FitF` and `Sim` for serving σ_t. Formally, $\delta_t \doteq 1_{\sigma_t \notin \hat{C}_t} - 1_{\sigma_t \notin C_t}$.

Define $\mathcal{H}_x^y$ for $x \in \mathbb{Z}$, $y \in \mathbb{Z}$ to be the set of rounds $t \in [T]$ where the $d_{t+1} - d_t = x$ and $\delta_t = y$. Let $\mathcal{H}_x = \bigcup_y \mathcal{H}_x^y$ and $\mathcal{H}^y = \bigcup_x \mathcal{H}_x^y$.

For a round $t \in [T]$, we say that t is a *troublemaker* if and only if t satisfies

$$\left(\hat{e}_t \neq \perp\right) \wedge \left(\hat{e}_t \in I_t\right) \wedge A_t(\hat{e}_t) \in [T] \wedge \left(\hat{e}_t \in C_{A_t(\hat{e}_t)}\right) \wedge \left(e_t = \perp \vee e_t \notin I_t\right). \tag{5}$$

The set of troublemaker rounds is denoted by Γ. For any troublemaker $\gamma \in \Gamma$ and any round $t \in (\gamma, T]$, we say γ is *active* at t if $t < A_\gamma(\hat{e}_\gamma)$. The set of troublemakers that are active at t is denoted by $\Gamma_t \subseteq \Gamma \cap [t-1]$. The *active period* $[\gamma + 1, A_\gamma(\hat{e}_\gamma) - 1]$ of γ is denoted by θ_γ.

Preliminary results. The following results are directly inferred from the definitions above.

▶ **Lemma 7.** *For any round t and any page i, the following properties are satisfied.*

- *If $R_t(i) = -1$ and $i \neq \sigma_t$, then $\hat{a}_t(i) = Z + 1$, and vice versa.*
- *The equality $\hat{a}_t(i) = A_t(i)$ holds if $a_r = A_r(\sigma_r)$, where $r = R_t(i)$.*

Proof. The first statement can be proved inductively with using Eq. (3). The first statement implies that if $\hat{a}_t(i) = A_t(i)$, then $R_t(i) \neq -1$. Therefore, the second statement can be inferred from Lemma 6. ◀

▶ **Lemma 8.** $|\{t \mid \hat{e}_t \neq \perp \wedge R_t(\hat{e}_t) = -1\}| \leq k$.

Proof. For each round t and each page $i \in \hat{C}_t$, the equality $R_t(i) = -1$ holds only if $i \in \hat{C}_1$. The initial cache profile $\hat{C}_1$ contains k different pages, and for each page $i \in \hat{C}_1$, if there exist two rounds t, t' with $t < t'$ and $\hat{e}_t = \hat{e}_{t'} = i$, then $R_{t'}(i) \geq t \neq -1$. This implies that there are at most k rounds t with $R_t(\hat{e}_t) = -1$. ◀

▶ **Lemma 9.** $|\{t \mid R_t(\sigma_t) = -1 \wedge t \in \mathcal{H}_0^1\}| \leq k$

Proof. For a round $t \in \mathcal{H}_0^1$, we have $\sigma_t \in C_t$. Since $R_t(\sigma_t) = -1$, we have $\sigma_t \in C_1$. Then this lemma can be proved in a similar way with Lemma 8. ◀

▶ **Lemma 10.** *For every round t and any troublemaker $\gamma \in \Gamma_t$, we have (1) $\hat{e}_\gamma \in C_t \setminus \hat{C}_t$, and (2) $\hat{e}_\gamma \neq \hat{e}_{\gamma'}$ for any troublemaker $\gamma' \in \Gamma_t$ with $\gamma \neq \gamma'$.*

Proof. The first statement is directly inferred from the definition of active troublemakers. Now consider the second statement. Without loss of generality, we assume that $\gamma < \gamma'$. Then the first statement shows that $\hat{e}_\gamma \notin \hat{C}_{\gamma'}$, which means that `Sim` cannot evict $\hat{e}_\gamma$ at γ'. ◀

3.2 Reducing Cost Analysis to Troublemaker Counting

▶ **Lemma 11.** *For each round $t \in [T]$, we have $d_{t+1} - d_t \in \{-1, 0, 1\}$ and $\delta_t \in \{-1, 0, 1\}$.*

The proof of Lemma 11 is deferred to the full version [11]. This lemma implies that for the sets $\mathcal{H}_x^y$, we only need to consider the parameters $x, y \in \{-1, 0, 1\}$. The following result on $\mathcal{H}_x^y$ can be inferred from the proof of Lemma 11.

▶ **Lemma 12.** *It holds that $\mathcal{H}_1 = \mathcal{H}_1^0 = \{t \mid e_t \neq \perp \bigwedge \hat{e}_t \neq \perp \bigwedge e_t \neq \hat{e}_t \bigwedge \hat{e}_t \in I_t \bigwedge e_t \in I_t\}$.*

Lemma 12 implies that $\mathcal{H}_1^1 = \emptyset$, therefore, cost(`Sim`) − `OPT` can be bounded by $|\mathcal{H}^1| - |\mathcal{H}^{-1}| = |\mathcal{H}_0^1| + |\mathcal{H}_{-1}^1| - |\mathcal{H}^{-1}|$.

▶ **Lemma 13.** $|\mathcal{H}_{-1}| \leq |\mathcal{H}_1| + k$.

Proof. By definition, $\mathcal{H}_{-1}$ is the set of rounds t with $d_{t+1} - d_t < 0$, and $\mathcal{H}_1$ is the set of rounds t with $d_{t+1} - d_t > 0$. Since $d_1 \leq k$ and $d_t \geq 0$ for every $t \in [T+1]$, this proposition holds. ◀

Lemma 13 allows us to bound $|\mathcal{H}_{-1}^1|$ with $|\mathcal{H}_1|$. The following result follows from the mechanisms of `FitF` and `Sim` in choosing the page for eviction when cache miss happens.

▶ **Lemma 14.** *For each round $t \in \mathcal{H}_1$, we have $A_t(\hat{e}_t) < A_t(e_t)$ and $\hat{a}_t(e_t) \leq \hat{a}_t(\hat{e}_t)$.*

For each round $t \in \mathcal{H}_1$, we say that t *blames* another round t' specified as follows. Let $r = R_t(\hat{e}_t)$ and $r' = R_t(e_t)$, then the round blamed by t is

$$t' = \begin{cases} r & \text{if } (r \neq -1) \wedge (a_r \neq A_r(\sigma_r)) \\ r' & \text{if } \big(r = -1 \vee a_r = A_r(\sigma_r)\big) \wedge \big(r' \neq -1 \wedge a_{r'} \neq A_{r'}(\sigma_{r'})\big) \\ -1 & \text{otherwise} \end{cases} .$$

▶ **Lemma 15.** *For each $t \in \mathcal{H}_1$, let t' be the round blamed by t. If $t' = -1$, then $R_t(\hat{e}_t) = -1$. If $R_t(\hat{e}_t) \neq -1$, then there exists a round t'' such that $\{t', t''\} \in$ `INV`.*

Proof. For the first claim, suppose on the contrary $r = R_t(\hat{e}_t) \neq -1$. Now consider two cases, $r' \neq -1$ and $r' = -1$, where $r' = R_t(e_t)$.

- $r' \neq -1$: Since $t' = -1$, in such a case we have $a_r = A_r(\sigma_r)$ and $a_{r'} = A_{r'}(\sigma_{r'})$. By Lemma 7, it holds that $\hat{a}_t(\hat{e}_t) = a_r = A_r(\sigma_r) = A_t(\hat{e}_t)$. Similarly, we have $\hat{a}_t(e_t) = A_t(e_t)$. This conflicts with Lemma 14.
- $r' = -1$: By Lemma 7, in such a case we have $\hat{a}_t(e_t) = Z + 1$, because $e_t \neq \sigma_t$. As it still holds that $\hat{a}_t(\hat{e}_t) = A_t(\hat{e}_t) < Z$, we have $\hat{a}_t(\hat{e}_t) < \hat{a}_t(e_t)$, which conflicts with Lemma 14.

For the second claim, since $R_t(\hat{e}_t) \neq -1$, Lemma 7 indicates that $\hat{a}_t(\hat{e}_t) \leq Z$. Now consider the following two cases.

- $\hat{a}_t(\hat{e}_t) < Z$: By Lemma 14, in such a case we also have $\hat{a}_t(e_t) < Z$. It can be inferred from Eq. (3) that $\hat{a}_t(\hat{e}_t) = a_r$ and $\hat{a}_t(e_t) = a_{r'}$. Notice that the second equality holds because $\hat{a}_t(e_t) < Z$ means that $r' \neq -1$. Then by Lemma 14, we have $a_{r'} = \hat{a}_t(e_t) \leq \hat{a}_t(\hat{e}_t) = a_r$ and $A_r(\sigma_r) = A_t(\hat{e}_t) < A_t(e_t) = A_{r'}(\sigma_{r'})$. This means that the pair $\{r, r'\}$ is an inversion. Since $R_t(\hat{e}_t) = -1$, we have either $t' = r$ or $t' = r'$. By taking

$$t'' = \begin{cases} r' & \text{if } t' = r \\ r & \text{if } t' = r' \end{cases},$$

this claim is proved.
- $\hat{a}_t(\hat{e}_t) = Z$: Let t_1 be the first round in $(r, t]$ so that $\hat{a}_{t_1}(\hat{e}_t) = Z$. Then Eq. (3) indicates that $\hat{a}_{t_1-1}(\sigma_{t_1}) < Z$. By Lemma 7, we have $r_1 = R_{t_1}(\sigma_{t_1}) \neq -1$. Then, $A_{r_1}(\sigma_{r_1}) = t_1 < A_t(\hat{e}_t) = A_r(\sigma_r)$. Moreover, $\hat{a}_{t_1-1}(\sigma_r) \leq \hat{a}_{t_1-1}(\sigma_{t_1}) < Z$ means that $a_r = \hat{a}_{t_1-1}(\sigma_r)$ and $a_{r_1} = \hat{a}_{t_1-1}(\sigma_{t_1})$. Therefore, the pair $\{r_1, r\}$ is an inversion. Then this claim is established if $t' = r$. This equation holds because (1) $r = R_t(\hat{e}_t) \neq -1$, and (2) Lemma 7 implies that $A_r(\sigma_r) \neq a_r$, because otherwise $\hat{a}_t(\hat{e}_t) = A_t(\hat{e}_t) < Z$.

This completes the proof. ◀

▶ **Lemma 16.** *For each round $t' \neq -1$, it can be blamed by at most two rounds in $\mathcal{H}_1$.*

Proof. Suppose that t' is blamed by $t \in \mathcal{H}_1$ such that $t' = R_t(e_t)$. Then for any round $t'' \in \mathcal{H}_1$ with $t' < t'' < t$, it cannot blame t' by taking $t' = R_{t''}(e_{t''})$. This is because if there exists such a round t'', then by definition we have $e_t = e_{t''}$. In such a case, there must exist a round $\tilde{t} \in (t'', t)$ with $\sigma_{\tilde{t}} = e_t$, otherwise $e_t \notin C_t$. This conflicts with the definition that $R_t(e_t)$ is the last round before t when e_t is requested. For any $t'' \in \mathcal{H}_1$ with $t'' > t$, it cannot blame t' by taking $t' = R_{t''}(e_{t''})$, either. Still, if $e_t = e_{t''}$, there must exist a round $\tilde{t} \in (t, t'')$ with $\sigma_{\tilde{t}} = e_t$. In such a case, $R_{t''}(e_{t''}) \geq \tilde{t} > t > t'$. The case $t' = R_t(\hat{e}_t)$ is symmetric with the case above. ◀

▶ **Lemma 17.** *It holds that $|\mathcal{H}_1| \leq 2 \cdot \eta + k$ and $|\mathcal{H}^1_{-1}| \leq 2 \cdot \eta + 2k$.*

Proof. The statement $|\mathcal{H}_1| \leq 2 \cdot \eta + k$ follows from Lemma 8, Lemma 15 and Lemma 16. The statement $|\mathcal{H}^1_{-1}| \leq 2 \cdot \eta + 2k$ then follows from Lemma 13. ◀

Next, we analyze $|\mathcal{H}^1_0|$ with the notion of troublemakers defined in Section 3.1. In particular, for two rounds t, t' with $t' < t$, we say t' is the *parent* of t and t is the *child* of t' if $t \in \mathcal{H}^1_0$ and $t' = \max\{t'' < t \mid \sigma_t \in \hat{C}_{t''}\}$.

▶ **Lemma 18.** *Every round $t \in \mathcal{H}^1_0 \setminus \{\tilde{t} \mid R_{\tilde{t}}(\sigma_{\tilde{t}}) = -1\}$ has one parent $t' \in \Gamma \cup \mathcal{H}_1$, and any round t' has at most one child.*

Proof. Since $t \notin \{\tilde{t} \mid R_{\tilde{t}}(\sigma_{\tilde{t}}) = -1\}$, the page σ_t is requested at round $r = R_t(i) \in [1, t)$, which gives that $\sigma_t \in \hat{C}_{r+1}$. As $t \in \mathcal{H}_0^1$, we know that $\sigma_t \notin \hat{C}_t$. Therefore, there must exist a round $t'' \in [r+1, t)$ with $\sigma_t = \hat{e}_{t''} \in \hat{C}_{t''}$. Since $\{t'' \mid t'' < t \wedge \sigma_t \in \hat{C}_{t''}\} \neq \emptyset$, the existence of the parent round of t is ensured.

For the parent t' of round t, we know that $\hat{e}_{t'} = \sigma_t$, because $\sigma_t \in \hat{C}_{t'} \setminus \hat{C}_{t'+1}$. Then we have $\hat{e}_{t'} \neq \perp$ and $\hat{e}_{t'} \in I_{t'}$, where the second equality holds because $\sigma_t \in C_t$ and $\sigma_{t''} \neq \sigma_t$ for every $t'' \in [t', t-1]$. Then by the definitions of the parent round and $\mathcal{H}_0^1$, we have $A_{t'}(\hat{e}_{t'}) = t \in [T]$ and $\hat{e}_{t'} \in C_t$, which means that $\hat{e}_{t'} \in C_{A_{t'}(\hat{e}_{t'})}$. Therefore, $t' \in \Gamma$ if $e_{t'} = \perp$ or $e_{t'} \notin I_{t'}$. If $e_{t'} \neq \perp$ and $e_{t'} \in I_t$, then we have $e_{t'} \neq \hat{e}_{t'}$, because otherwise $\hat{e}_{t'} \notin C_t$. By Lemma 12, in such a case we have $t' \in \mathcal{H}_1$.

It remains to prove that any round t' has at most one child. Suppose that t' has two children t_1, t_2 with $t_1 < t_2$. In such a case, $\sigma_{t_2} = \hat{e}_{t'} = \sigma_{t_1} \in \hat{C}_{t_1+1}$, which conflicts with the definition of the parent round. ◀

Putting Lemma 9, Lemma 17, and Lemma 18 together gives the following result.

▶ **Theorem 19.** $\mathrm{cost}(\texttt{Sim}) - \texttt{OPT} \leq 4\eta + 4k + |\Gamma| - |\mathcal{H}^{-1}|$.

We defer the proof of Theorem 19 to the full version [11]. The upper bound on $|\Gamma| - |\mathcal{H}^{-1}|$ is studied in the next subsection.

3.3 Labeling for Troublemakers

From the high level, the analysis in this part on $|\Gamma| - |\mathcal{H}^{-1}|$ is done by showing that each troublemaker γ either can be mapped a distinct round in $\mathcal{H}^{-1}$, or can be mapped to a round $t < \gamma$ that has a prediction error. We specify the mappings with a procedure called `Labeling`, which is designed to avoid mapping too many troublemakers to a single prediction error. Notice that `Labeling` is only used in the analysis, while the paging algorithm is unaware of the output generated by `Labeling`.

Procedure `Labeling` takes $\langle \{A_t(i)\}_{t\in[T],i\in[n]}, \{a_t\}_{t\in[T]} \rangle$ as the input, which implicitly encodes the operations of `FitF` and `Sim`, and for each troublemaker $\gamma \in \Gamma$, `Labeling` outputs a labeling function $\lambda_\gamma : \theta_\gamma \mapsto \big([n] \cup \perp\big)$, which maps each round in the active period θ_γ of γ to either a page i or an empty value. For each $\gamma \in \Gamma$ and each round t in the active period θ_γ with $\lambda_\gamma(t) \neq \perp$, we say that page $i = \lambda_\gamma(t)$ is labelled by γ. Procedure `Labeling` is presented in Algorithm 1 with notions defined as follows.

$$\forall t \in [T]: \qquad \mathcal{L}_t \doteq \bigcup_{\gamma \in \Gamma_t} \lambda_\gamma(t), \quad \text{and} \quad \Phi_t \doteq \hat{C}_t \setminus \Big(C_t \cup \mathcal{L}_t\Big),$$

$$\forall t \in [2, T]: \qquad \Psi_t \doteq \hat{C}_t \setminus \Big(C_t \cup \mathcal{L}_{t-1}\Big).$$

Briefly speaking, for every $\gamma \in \Gamma$, `Labeling` picks an arbitrary page $i = \hat{\delta}_\gamma$ from Φ_γ and labels i with γ for the first round in the active period of γ, which means setting $\lambda_\gamma(\gamma+1) = i$. For convenience, the NAT of i after γ is denoted by τ_γ. Then we consider the following cases.

- $\tau_\gamma > A_\gamma(\hat{e}_\gamma)$: Then the label on $\hat{\delta}_\gamma$ is kept throughout the active period θ_γ of γ.
- $\tau_\gamma < A_\gamma(\hat{e}_\gamma)$ and $\hat{\delta}_\gamma \in \hat{C}_{\tau_\gamma}$: This means that $\hat{\delta}_\gamma$ is not evicted by `Sim` before its NAT after γ. In such a case, the label on $\hat{\delta}_\gamma$ is kept until the last round before its NAT.
- $\tau_\gamma < A_\gamma(\hat{e}_\gamma)$ and $\hat{\delta}_\gamma \notin \hat{C}_{\tau_\gamma}$: In such a case, a labelled page is evicted by `Sim` before its NAT after γ. For each round t with such an eviction, we label a new page at round $t+1$ in Ψ_{t+1} with γ. We stop labelling new pages either when the labelled page is requested, or the NAT of the previous labelled page after the previous round is less than the NAT of the current labelled page.

Algorithm 1 Procedure `Labeling`.

Input: $\{A_t(i)\}_{t \in [T], i \in [n]}$, $\{a_t\}_{t \in [T]}$
Output: $\{\lambda_\gamma\}_{\gamma \in \Gamma}$

for *each* $\gamma \in \Gamma$ **do**
  Pick an arbitrary element $i \in \Phi_\gamma$ and set $\hat{\delta}_\gamma = i$;
  Set $\lambda_\gamma(\gamma + 1) = \hat{\delta}_\gamma$ and $\tau_\gamma = A_\gamma(\hat{\delta}_\gamma)$;
  if $\tau_\gamma > A_\gamma(\hat{e}_\gamma)$ **then**
    Set $\lambda_\gamma(t) = \hat{\delta}_\gamma$ for all the remaining rounds t in $\theta\gamma$;
  else
    Set $t = \gamma + 2$;
    while $t < A_\gamma(\hat{e}_\gamma)$ **do**
      if $\lambda_\gamma(t-1) = \sigma_{t-1}$ **then**
        break;
      if $\lambda_\gamma(t-1) \neq \hat{e}_{t-1}$ **then**
        Set $\lambda_\gamma(t) = \lambda_\gamma(t-1)$;
      else
        Pick an arbitrary element i from Ψ_t and set $\lambda_\gamma(t) = i$;
        if $A_{t-1}(\lambda_\gamma(t)) > A_{t-1}(\lambda_\gamma(t-1))$ **then**
          Set $\lambda_\gamma(t') = i$ for every round $t' \in [t+1, \min\{A_\gamma(\hat{e}_\gamma), A_t(i)\})$;
          Set $t = \min\{A_\gamma(\hat{e}_\gamma), A_t(i)\} - 1$;
          break;
      Set $t = t + 1$;
    Set $\lambda_\gamma(t') = \perp$ for every $t' \in [t, A_\gamma(\hat{e}_\gamma))$;

Before describing how Procedure `Labeling` is applied to map each troublemaker to a round with a prediction error or a round in $\mathcal{H}^{-1}$, we first prove that this procedure is consistent by showing that Φ_t (resp. Ψ_t) is not empty whenever we need to find a new page to label from Φ_t (resp. Ψ_t). For every troublemaker round $t \in \Gamma$, define

$$\zeta_t = \begin{cases} \sigma_t & \text{if } e_t = \perp \\ e_t & \text{otherwise} \end{cases} .$$

Then we have the following results.

▶ **Lemma 20.** *For each troublemaker round $t \in \Gamma$, we have (1) $\zeta_t \neq \perp$, (2) $\zeta_t \in C_t \setminus \hat{C}_t$, and (3) for any $\gamma \in \Gamma_t$, we have $\zeta_t \neq \hat{e}_\gamma$.*

Proof. Claim (1) and (2) are obvious. Now consider claim (3). Since $\gamma < t < A_\gamma(\hat{e}_\gamma)$, $\sigma_t \neq \hat{e}_\gamma$, because otherwise $t = A_\gamma(\hat{e}_\gamma)$. By the definition of troublemakers, $\hat{e}_\gamma \in C_{A_\gamma(\hat{e}_\gamma)}$, so for any $t \in [\gamma + 1, A_\gamma(\hat{e}_\gamma) - 1]$, it holds that $e_t \neq \hat{e}_\gamma$. ◀

▶ **Lemma 21.** *For each troublemaker round $t \in \Gamma$, it holds that $|\Phi_t| \geq 1$.*

Proof. By Lemma 10 and Lemma 20, we have $C_t \setminus \hat{C}_t \supseteq \{\hat{e}_\gamma\}_{\gamma \in \Gamma_t} \cup \{\zeta_t\}$. Still by Lemma 20, it follows that $\zeta_t \neq \hat{e}_\gamma$ for every $\gamma \in \Gamma_t$, then $|C_t \setminus \hat{C}_t| \geq \left|\{\hat{e}_\gamma\}_{\gamma \in \Gamma_t}\right| + 1 = \left|\Gamma_t\right| + 1$. Because $|C_t| = |\hat{C}_t|$, we have $|\hat{C}_t \setminus C_t| \geq \left|\Gamma_t\right| + 1$. Since for every γ with $t \in \theta_\gamma$, we have $\gamma \in \Gamma_t$, then it always holds that $|\mathcal{L}_t| \leq |\Gamma_t|$. This finishes the proof. ◀

▶ **Lemma 22.** *For each round $t \in [T]$, we have $C_t \cap \mathcal{L}_t = \emptyset$.*

Proof. This proposition is true because for any round t when we find a new page i to label, $i \notin C_t$, and the label on any page i is cancelled before i is requested. ◀

▶ **Lemma 23.** *For every $\gamma \in \Gamma$ and every $t \in [\gamma + 2, A_\gamma(\hat{e}_\gamma) - 1]$, if $\hat{e}_{t-1} = \lambda_\gamma(t-1)$, then $\Psi_t \neq \emptyset$.*

Proof. First, consider the case where $\Phi_{t-1} \neq \emptyset$. For each page $i \in \Phi_{t-1}$, we know $i \neq \hat{e}_{t-1}$ because $i \notin \mathcal{L}_{t-1}$ while $\hat{e}_{t-1} = \lambda_\gamma(t-1) \in \mathcal{L}_{t-1}$. This gives $i \in \hat{C}_t$. Moreover, we have $i \neq \sigma_{t-1}$ because $\hat{e}_{t-1} \neq \perp$. This gives $i \notin C_t$. Therefore, $i \in \hat{C}_t \setminus (C_t \cup \mathcal{L}_{t-1}) = \Psi_t$.

Now consider Φ_{t-1} is empty. In such a case, $|\hat{C}_{t-1} \setminus C_{t-1}| = |\mathcal{L}_{t-1}|$. Lemma 10 indicates that for every $\gamma' \in \Gamma_{t'}$ with $t' \in \theta_{\gamma'}$, we have $\hat{e}_{\gamma'} \in C_{t'} - \hat{C}_{t'}$. Then $|\hat{C}_{t-1} \setminus C_{t-1}| = |\mathcal{L}_{t-1}|$ implies that $C_{t-1} \setminus \hat{C}_{t-1} = \{\hat{e}_{\gamma'}\}_{\gamma' \in \Gamma_{t-1}}$. By the definition of active troublemakers, $\sigma_{t-1} \notin C_{t-1} - \hat{C}_{t-1}$. Also, we have $\sigma_{t-1} \notin I_{t-1} = C_{t-1} \cap \hat{C}_{t-1}$, because $\hat{e}_{t-1} \neq \perp$. Therefore, $\sigma_{t-1} \notin C_{t-1}$, which means that $e_{t-1} \neq \perp$. Still by the definition of the troublemakers, we have $e_{t-1} \notin \{\hat{e}_{\gamma'}\}_{\gamma' \in \Gamma_{t-1}} = C_{t-1} \setminus \hat{C}_{t-1}$. Thus, $e_{t-1} \in I_{t-1} \subseteq \hat{C}_{t-1}$. By Lemma 22, we have $e_{t-1} \notin \mathcal{L}_{t-1}$ and $e_{t-1} \neq \hat{e}_{t-1}$. Putting $e_{t-1} \neq \hat{e}_{t-1}$ and $e_{t-1} \in \hat{C}_{t-1}$ together, we get $e_{t-1} \in \hat{C}_t - C_t$. Therefore, $e_{t-1} \in \Psi_t$. ◀

Lemma 21 and Lemma 23 ensure the consistency of Procedure `Labeling`.

▶ **Lemma 24.** *For every round t and any page $i \in [n]$, there exists at most one troublemaker $\gamma \in \Gamma_t$ so that $\lambda_\gamma = i$.*

The proof of Lemma 24 is deferred to the full version [11]. It also gives the following byproduct.

▶ **Lemma 25.** *For each round $t \in [2, T]$, we have $|\mathcal{L}_t \setminus \mathcal{L}_{t-1}| \leq 1$.*

Let t be an arbitrary round with $\hat{e}_t \neq \perp$. Suppose that there exists a page $i \in \hat{C}_t$ satisfying $A_t(i) > A_t(\hat{e}_t)$, $i \notin \mathcal{L}_t$, and $i \in \mathcal{L}_{t'}$ for every $t' \in [t+1, \min\{A_t(\hat{e}_t), t^\circ\}]$, where

$$t^\circ = \begin{cases} \min\{t'' \mid t'' > t \land \hat{e}_{t''} = i\} & \text{if } \exists t'' \in (t+1, T] \text{ s.t. } \hat{e}_{t''} = i \\ \infty & \text{otherwise} \end{cases} . \tag{6}$$

Lemma 25 implies that such a page i is unique if it exists. In such a case, we say that the page i is the *competitor* of $\hat{e}_t$, and the round t has an *abettor* round t^* specified as follows. Let $r = R_t(\hat{e}_t)$ and $r' = R_t(i)$, then the abettor of t is

$$t^* = \begin{cases} r & \text{if } r \neq -1 \land A_r(\sigma_r) \neq a_r \\ r' & \text{if } \big(r = -1 \lor A_r(\sigma_r) = a_r\big) \land \big(r' \neq -1 \land A_{r'}(\sigma_{r'}) \neq a_{r'}\big) \\ -1 & \text{otherwise} \end{cases} . \tag{7}$$

▶ **Lemma 26.** *For any round $t^* \in [T]$, the number of rounds in $\big\{t \,|\, \hat{e}_t \neq \perp \land R_t(\hat{e}_t) \neq -1 \land t^* \text{ is the abettor of } t\big\}$ is at most two.*

Proof. It can be proved in a similar way with Lemma 16 that $\big\{t \,|\, \hat{e}_t \neq \perp \land R_t(\hat{e}_t) \neq -1 \land t^* \text{ is the abettor of } t \land t^* = R_t(\hat{e}_t)\big\}$ contains at most one round. It remains to prove that $\big\{t \,|\, \hat{e}_t \neq \perp \land R_t(\hat{e}_t) \neq -1 \land t^* \text{ is the abettor of } t \land t^* = R_t(i)\big\}$ contains at most a single round, where i is the competitor of $\hat{e}_t$ as defined in Eq. (6). Notice that different from the case considered in the proof of Lemma 16, the page i may not be evicted by `FitF`. Therefore, we need to utilize the properties of procedure `Labeling` to prove the uniqueness of the round t which satisfies $t^* = R_t(i)$.

▷ Claim 27. If a round t has an abettor $t^* = R_t(i)$ with i being the competitor of $\hat{e}_t$ and $r = R_t(\hat{e}_t) \neq -1$, then it holds that $\hat{a}_t(i) < Z$ and $\hat{a}_{t'}(i) = Z$ for any round $t' \in [A_t(\hat{e}_t), A_t(i))$.

Proof. Since $t^* = R_t(i)$ and $r \neq -1$, we have $A_r(\sigma_r) = a_r$, which by Lemma 7 gives $A_t(\hat{e}_t) = \hat{a}_t(\hat{e}_t) < Z$. Following Lemma 14, we have $\hat{a}_t(i) \leq A_t(\hat{e}_t)$ and $\hat{a}_t(i) < Z$. If there exists a round $t'' \in (t, A_t(\hat{e}_t))$ so that $\hat{a}_{t''}(i) = Z$, then it follows from Eq. (3) that for any round $t' \in [A_t(\hat{e}_t), A_t(i))$, it holds that $\hat{a}_{t'}(i) = Z$, which means that this proposition holds. We proceed to prove that $\hat{a}_{A_t(\hat{e}_t)}(i) = Z$ if $\hat{a}_{t''}(i) \neq Z$ holds for every round $t'' \in (t, A_t(\hat{e}_t))$. In such a case, it can be inferred from Eq. (3) that $\hat{a}_{t''}(i) = \hat{a}_t(i)$. Since $A_r(\sigma_r) = a_r$, Lemma 7 indicates that $\hat{a}_{t''}(\hat{e}_t) = A_{t''}(\hat{e}_t) = A_t(\hat{e}_t) = \hat{a}_t(\hat{e}_t)$ and $\hat{a}_{t''}(\hat{e}_t) < Z$ hold for any round $t'' \in (t, A_t(\hat{e}_t))$. Combining $\hat{a}_{t''}(\hat{e}_t) = \hat{a}_t(\hat{e}_t)$ with $\hat{a}_{t''}(i) = \hat{a}_t(i)$ gives $\hat{a}_{t''}(\hat{e}_t) \geq \hat{a}_{t''}(i)$. Therefore, $\hat{a}_{A_t(\hat{e}_t)-1}(i) \leq \hat{a}_{A_t(\hat{e}_t)-1}(\hat{e}_t) < Z$. Moreover, we have $\hat{a}_{A_t(\hat{e}_t)-1}(i) < A_t(\hat{e}_t)$ because $A_t(\hat{e}_t) \geq \hat{a}_t(i) = \hat{a}_{t''}(i)$ holds for every $t'' \in (t, A_t(\hat{e}_t))$. By Eq. (3), the conditions for $\hat{a}_{A_t(\hat{e}_t)}(i) = Z$ are all satisfied. Thus, the equality $\hat{a}_{t'}(i) = Z$ holds for any round $t' \in [A_t(\hat{e}_t), A_t(i))$. ◁

For any round $t' \in (t^*, t)$, the round t^* cannot be the abettor of t' with taking $t^* = R_{t'}(i)$, because otherwise,

- if $A_{t'}(\hat{e}_{t'}) \geq t$, then by the definition of abettors, we have $i \in \mathcal{L}_t$, which conflicts with the requirement in the definition of abettors; else
- if $A_{t'}(\hat{e}_{t'}) < t$, then we get $\hat{a}_t(i) = Z$ with using the second statement in Claim 27, which conflicts with the first statement in Claim 27.

Therefore, this proposition holds. ◀

The following result can be proved by following the same line of arguments with the proof of Lemma 15.

▶ **Lemma 28.** *Let t be an arbitrary round that has an abettor t^*. If $t^* = -1$, then $R_t(\hat{e}_t) = -1$. If $R_t(\hat{e}_t) \neq -1$, then there exists a round t' such that $\{t^*, t'\} \in \texttt{INV}$.*

▶ Remark 29. Notice that the statement of Lemma 28 is consistent because for any round t having an abettor, by definition we have $\hat{e}_t \neq \perp$.

For an arbitrary round t, if there exists a page i in $\hat{C}_t$ satisfies (1) $A_t(i) \leq T$, (2) $i \notin \mathcal{L}_t$ and (3) $i \in \mathcal{L}_{t'}$ for every $t' \in [t+1, A_t(i)]$, we say that $t^\triangle = A_t(i)$ is the *savior* of t.

▶ **Lemma 30.** *If a round t has a savior $t^\triangle$, then (1) $t^\triangle \in \mathcal{H}^{-1}$, and (2) for any $t^\triangle \in \mathcal{H}^{-1}$, it is the savior of at most one step t.*

Proof. The first claim follows from the definition of $\mathcal{H}^{-1}$. For any round $t' \in [t+1, t^\triangle]$, $t^\triangle$ is not the savior of t', because $\sigma_{t^\triangle} \in \mathcal{L}_{t'}$. Therefore, the second claim holds. ◀

▶ **Theorem 31.** *$|\Gamma| - |\mathcal{H}^{-1}| \leq 2 \cdot \eta + k$.*

Proof. The main idea of this proof is to show that each troublemaker can be mapped to a distinct *broker* round, and each broker either has an abettor or has a savior. In particular, we classify the troublemakers $\gamma \in \Gamma$ into the following three categories.

1. $\{\gamma \in \Gamma \mid \tau_\gamma > A_\gamma(\hat{e}_\gamma)\}$: Here, the troublemaker γ as a round has an abettor t^*, because $\hat{\delta}_\gamma \in \mathcal{L}_t$ for every step $t \in [\gamma+1, \min\{A_\gamma(\hat{e}_\gamma), t^\circ\}]$ where t° is defined in the same way with Eq. (6). In such a case, we say γ is the broker of itself.
2. $\{\gamma \in \Gamma \mid \tau_\gamma < A_\gamma(\hat{e}_\gamma) \wedge \hat{\delta}_\gamma \in \hat{C}_{\tau_\gamma}\}$: Now the troublemaker γ as a round has a savior τ_γ, because by the definition of the troublemaker, $A_\gamma(\hat{e}_\gamma) \leq T$, which gives $\tau_\gamma \leq T$. Moreover, it holds for every round $t \in [\gamma+1, A_\gamma(\hat{\delta}_\gamma)]$ that $\hat{\delta}_\gamma \in \mathcal{L}_t$. The broker for such a troublemaker γ is also itself.

3. $\{\gamma \in \Gamma \mid \tau_\gamma < A_\gamma(\hat{e}_\gamma) \wedge \hat{\delta}_\gamma \notin \hat{C}_{\tau_\gamma}\}$: In such a case, let i be the last page labelled by γ and t be the first round with $\lambda_\gamma(t) = i$. Since $\hat{\delta}_\gamma \notin \hat{C}_{\tau_\gamma}$, we have $t-1 \in \theta_\gamma$. Let $i' = \lambda_\gamma(t-1)$, then we have $i' \in \hat{C}_{t-1}$ because $\hat{e}_{t-1} = i'$. Also, we have $i \in \hat{C}_{t-1}$, because i is chosen from $\Psi_t \subseteq \hat{C}_t \setminus C_t$ and $i \neq \hat{e}_{t-1}$. Now consider two subcases.
   - $A_{t-1}(i') < A_{t-1}(i)$: In such a case, the round $t-1$ has an abettor t^*, because we have $i \in \mathcal{L}_{t'}$ for every $t' \in [t, \min\{A_{t-1}(i), \tau_\gamma, t^\circ\}]$, which satisfies the requirement in the definition of abettors because $\tau_\gamma > A_{t-1}(i')$.
   - $A_{t-1}(i') > A_{t-1}(i)$: It can be proved inductively that $A_{t-1}(i) < \tau_\gamma$, which implies that $A_{t-1}(i) \leq T$. By definition, it follows that $t-1$ has a savior $t^\triangle = A_{t-1}(i)$.

   The round $t-1$ is said to be the *broker* of the troublemaker γ. Lemma 24 ensures that the round $t-1$ cannot be the broker of two different troublemakers, because $\hat{e}_{t-1} = \lambda_\gamma(t-1)$. Moreover, by Lemma 25, the broker $t-1$ is not a troublemaker.

To sum up, each troublemaker γ can be mapped to a distinct broker t, and each broker t either has an abettor or has a savior. Then by Lemma 26, Lemma 28 and Lemma 30, this theorem holds. ◀

The following result is obtained by combining Theorem 19 and Theorem 31.

▶ **Theorem 32.** *It follows that* $\mathrm{cost}(\mathtt{Sim}) - \mathtt{OPT} \leq 6\eta + 5k$.

Because the cumulative prediction error is assumed to satisfy $\eta \in o(T)$, we have $\mathrm{cost}(\mathtt{Sim}) - \mathtt{OPT} \in o(T)$. Therefore, `Sim` has the vanishing regret when there is a single predictor.

4 Multiple NAT Predictors

This section extends the result obtained in Section 3 to the general case where there are $M > 1$ predictors making NAT predictions under the bandit access model. Our results on the full information access model are deferred to the full version [11].

For the bandit access model, in this part, we design an algorithm called *Sightless Chasing and Switching* (`S-C&S`) and prove that it has the vanishing regret.

The procedure of `S-C&S` is described in Algorithm 2. It is assumed that `S-C&S` is provided with blackbox accesses to the online algorithm *Implicitly Normalized Forecaster* (`INF`) [2] for the *Multiarmed Bandit Problem (MBP)* [3]. The MBP problem is an online problem defined over $\Upsilon \in \mathbb{Z}_{>0}$ rounds and a set X of arms. An oblivious adversary specifies a cost function $F_\upsilon : X \mapsto [0, 1]$ for each round $\upsilon \in [\Upsilon]$ that maps each arm $x \in X$ to a cost in $[0, 1]$. An algorithm for MBP needs to choose an arm x_υ at the beginning of each round $\upsilon \in [\Upsilon]$, and then the cost $F_\upsilon(x_\upsilon)$ incurred by the chosen arm x_υ is revealed to the algorithm. The objective of MBP is to minimize the cumulative cost incurred by the chosen arms $\{x_\upsilon\}_{\upsilon \in \Upsilon}$.

▶ **Theorem 33** ([2]). *The algorithm `INF` ensures that the chosen arms* $\{x_\upsilon\}_{\upsilon \in [\Upsilon]}$ *satisfy*

$$\sum_{\upsilon \in [\Upsilon]} F_\upsilon(x_\upsilon) - \min_{x^* \in X} \sum_{\upsilon \in [\Upsilon]} F_\upsilon(x^*) \in O\big(\sqrt{|X| \cdot \Upsilon}\big) .$$

Our algorithm `S-C&S` partitions the rounds into consecutive epochs of length $\tau \in \mathbb{Z}_{>0}$ and initializes `INF` by setting $\Upsilon = \lceil \frac{T}{\tau} \rceil$ and $X = [M]$, which means that each epoch in the online paging problem is mapped to a round in MBP, and each predictor is taken as an arm. The choice of the value for τ is discussed later. At the beginning of the first round $t_1^\upsilon = (\upsilon - 1)\tau + 1$ in each epoch $\upsilon \in [\Upsilon]$, `S-C&S` accesses `INF` to pick one predictor $j_{t_1^\upsilon}$. Then,

Algorithm 2 Algorithm S-C&S.

Input: $\{\sigma_t\}_{t\in[T]}$, MBP algorithm INF, initial cache profile $\hat{C}_1$
Output: $\{\hat{e}_t\}_{t\in[T]}$

Initialize the MBP algorithm INF with the number of rounds $\Upsilon = \lceil T/\tau \rceil$ and the set of arms $X = [M]$;
for *each round* $t \in [T]$ **do**
  if $t \bmod \tau = 1$ **then**
    Invoke INF to choose a predictor $j_t \in [M]$;
    Set $j = j_t$;
  else
    Set $j = j_{t'}$ with $t' = \lfloor t/\tau \rfloor \cdot \tau + 1$;
  Query the predictor j to obtain the prediction a_t^j;
  for *each page* $i \in [n]$ **do**
    if $i = \sigma_t$ **then**
      Set $\hat{a}_t(i) = a_t^j$;
    else if $t \bmod \tau = 1$ **then**
      Set $\hat{a}_t(i) = Z + 1$;
    else if $\hat{a}_{t-1}(i) = t \wedge i \neq \sigma_t \wedge \hat{a}_{t-1}(i) \leq \hat{a}_{t-1}(\sigma_t) < Z$ **then**
      Set $\hat{a}_t(i) = Z$;
    else
      Set $\hat{a}_t(i) = \hat{a}_{t-1}(i)$;
  if $\sigma_t \notin \hat{C}_t$ **then**
    Set $\hat{e}_t$ be the page $i \in C_t$ that maximizes $\hat{a}_t(i)$ with breaking ties arbitrarily;
    Update $\hat{C}_{t+1} = (\hat{C}_t \setminus \{\hat{e}_t\}) \cup \{\sigma_t\}$;
  else
    Set $\hat{e}_t = \perp$, and set $\hat{C}_{t+1} = \hat{C}_t$;
  if $t \bmod \tau = 0$ **then**
    Set $f = 0$;
    for *each round* $t' \in [t - \tau + 1, t]$ **do**
      if $(t' = t - \tau + 1) \vee (\hat{e}_{t'} \neq \perp) \vee \Big((\hat{e}_{t'} = \perp) \wedge (t' > t - \tau + 1) \wedge (\hat{a}_{t'-1}(\sigma_{t'}) = Z + 1)\Big)$ **then**
        Set $f = f + 1$;
    Send $\frac{f}{\tau}$ to INF as the cost incurred by $j_{t-\tau+1}$ in the epoch $\frac{t}{\tau}$;

S-C&S simulates the algorithm Sim, which is proposed in Section 3, throughout the epoch υ with taking t_1^υ as its initial round, $\hat{C}_{t_1^\upsilon}$ as its initial cache profile, and $j_{t_1^\upsilon}$ as the single predictor. At the end of the last round $t_\tau^\upsilon = \upsilon \cdot \tau$ in epoch υ, S-C&S sends

$$F_\upsilon(j_{t_1^\upsilon}) = \frac{1}{n}\left|\left\{t' \in [t_1^\upsilon, t_\tau^\upsilon] \,\middle|\, (t' = t_1^\upsilon) \vee (\hat{e}_{t'} \neq \perp) \vee \Big((\hat{e}_{t'} = \perp) \wedge (t' > t_1^\upsilon) \wedge (\hat{a}_{t'-1}(\sigma_{t'}) = Z+1)\Big)\right|\right. \quad (8)$$

to INF as the cost $F_\upsilon(j_{t_1})$ of choosing j_{t_1} for υ.

Notice that in MBP, the cost functions are generated by an oblivious adversary. We take this setting as a requirement that the cost function F_υ for each round υ in MBP should not depend on the arms chosen in the previous rounds $x_1, x_2, \ldots, x_{\upsilon-1}$. The following result shows that by feeding INF a cost that can be larger than the normalized cost that is actually incurred by S-C&S in the epoch, this requirement is satisfied.

▶ **Lemma 34.** *Let $F_\upsilon\left(j_{t_1^\upsilon}\middle|j_{t_1^1},\ldots,j_{t_1^{\upsilon-1}}\right)$ be the cost sent by* S-C&S *to* INF *at the end of an arbitrary epoch υ conditioned on the the predictors chosen for the previous epochs $j_{t_1^1},\ldots,j_{t_1^{\upsilon-1}}$. Then for any different sequence of predictors $\tilde{j}_{t_1^1},\ldots,\tilde{j}_{t_1^{\upsilon-1}}$, we have $F_\upsilon\left(j_{t_1^\upsilon}\middle|j_{t_1^1},\ldots,j_{t_1^{\upsilon-1}}\right) = F_\upsilon\left(j_{t_1^\upsilon}\middle|j'_{t_1^1},\ldots,j'_{t_1^{\upsilon-1}}\right)$.*

Proof. For the epoch υ, a round t is said to be *fresh* if for any earlier round $t' < t$ in υ, it holds that $\sigma_t \neq \sigma_{t'}$. We first consider the case where are at least k fresh rounds in the epoch, which means that at least k different pages are requested. Let t be the first round after the first k fresh rounds. We first consider the interval $[t_1^\upsilon, t-1]$. For each page $i \in [n]$ and each round $t' \in [t_1^\upsilon, t-1]$, we say i is *marked* at t' if there exists a round $t'' \in [t_1^\upsilon, t']$ with $\sigma_{t''} = i$, otherwise i is said to be *unmarked* at t'. Then it can be inferred from Lemma 7 that $\hat{a}_{t'}(i) = Z + 1$ if i is unmarked at t', and $\hat{a}_{t'}(i) \leq Z$ if i is marked at t'. Thus, there is no marked page getting evicted before t, and any round $t' \in [t_1^\upsilon, t-1]$ that satisfies $\hat{e}_{t'} \neq \perp$ must be a fresh round.

▷ Claim 35. For each round $t' \in [t_1^\upsilon, t-1]$, it is counted by Eq. (8) if and only if t' is fresh.

Proof. By definition, the round t_1^υ is a fresh round. Now consider a round $t' \in [t_1^\upsilon + 1, t-1]$. As mentioned above, if $\hat{e}_{t'} \neq \perp$, then t' is fresh. Also, if t' is fresh, then it can be inferred from Lemma 7 that $\hat{a}_{t'-1}(\sigma_{t'}) = Z + 1$. If t' is not fresh, which means that $\sigma_{t'}$ is marked, then it holds that $\hat{e}_{t'} = \perp$ and $\hat{a}_{t'-1}(\sigma_{t'}) \leq Z$. Therefore, this claim holds. ◁

Therefore, the contribution of the rounds in $[t_1^\upsilon, t-1]$ to $F_\upsilon(j_{t_1^\upsilon})$ is always $\frac{k}{n}$, which is independent of $j_{t_1}^1, \ldots, j_{t_1}^{\upsilon-1}$. A similar result can also be obtained when there are less than k fresh rounds in the epoch υ.

At the beginning of round t, $\hat{C}_t$ contains exactly the first k different pages required in the epoch υ, and for each page $i \in \hat{C}_t$, the remedy prediction $\hat{a}_t(i)$ is computed only based on $\{a_{t'}^{j_{t_1^\upsilon}}\}_{t' \in [t_1^\upsilon, t]}$ and $\{\sigma_{t'}\}_{t' \in [t_1^\upsilon, t]}$. Therefore, for any $t' \geq t$, the decision on $\hat{e}_{t'}$ is independent of the choices over $j_{t_1}^1, \ldots, j_{t_1}^{\upsilon-1}$.

Furthermore, since at round $t-1$, every page $i \in \hat{C}_{t-1}$ is marked, then for any round $t' \geq t$ with $\hat{e}_t = \perp$, the page $\sigma_{t'}$ has been requested at least once in the interval $[t_1^\upsilon, t'-1]$, which means that $\hat{a}_{t'-1}(\sigma_{t'}) \leq Z$. This observation is formally stated in the following claim.

▷ Claim 36. For any round $t' \geq t$, it is counted by Eq. (8) if and only if $\hat{e}_{t'} \neq \perp$.

Thus, the contribution of the rounds in $[t, \upsilon \cdot \tau]$ to $F_\upsilon(j_{t_1^\upsilon})$ does not depend on the previous epochs $\{j_{t_1}^1, \ldots, j_{t_1}^{\upsilon-1}\}$, either. ◀

For an epoch $\upsilon \in \Upsilon$ and a predictor $j \in [M]$ chosen for υ, let $\sharp\texttt{Evictions}_\upsilon(j) = |\{t \in [t_1^\upsilon, t_\tau^\upsilon] \mid \hat{e}_t \neq \perp\}|$. The following result is also obtained by combining Claim 35 with Claim 36.

▶ **Lemma 37.** *For each epoch υ, $\sharp\texttt{Evictions}_\upsilon(j) \leq \tau \cdot F_\upsilon(j) \leq \sharp\texttt{Evictions}_\upsilon(j) + k$.*

The following result is obtained by putting Theorem 32, Theorem 33, Lemma 34, and Lemma 37 together.

▶ **Theorem 38.** *By taking $\tau = \lfloor T^{\frac{1}{3}} \rfloor$, the regret of* S-C&S *is bounded by $O\big(kT^{\frac{2}{3}}\sqrt{M} + \eta^{\min}\big)$.*

The proof of this theorem is deferred to the full version [11]. By assumption, we have $\eta^{\min} \in o(T)$. Therefore, Theorem 38 implies that S-C&S has a vanishing regret.

Note that the result in this section cannot be obtained by using the results in [12] directly, because the algorithms proposed in [12] requires to know the cache profile of the algorithm Sim that follows each predictor, which is unavailable under the bandit access model.

References

1 Antonios Antoniadis, Christian Coester, Marek Eliás, Adam Polak, and Bertrand Simon. Online metric algorithms with untrusted predictions. In *Proceedings of Machine Learning and Systems*, ICML '20, pages 11463–11473. PMLR, 2020. URL: https://proceedings.icml.cc/static/paper_files/icml/2020/6657-Paper.pdf.

2 Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *COLT '09 - The 22nd Conference on Learning Theory*, Montreal, Quebec, Canada, June 2009. URL: http://www.cs.mcgill.ca/%7Ecolt2009/papers/022.pdf#page=1.

3 Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002. doi:10.1137/S0097539701398375.

4 Grant Ayers, Heiner Litz, Christos Kozyrakis, and Parthasarathy Ranganathan. Classifying memory access patterns for prefetching. In *Architectural Support for Programming Languages and Operating Systems*, ASPLOS '20, pages 513–526, Lausanne, Switzerland, March 2020. ACM. doi:10.1145/3373376.3378498.

5 Laszlo A. Belady. A study of replacement algorithms for virtual-storage computer. *IBM Systems Journal*, 5(2):78–101, 1966. doi:10.1147/sj.52.0078.

6 Avrim Blum and Carl Burch. On-line learning and the metrical task system problem. *Machine Learning*, 39(1):35–58, 2000. doi:10.1023/A:1007621832648.

7 Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998.

8 Allan Borodin, Nathan Linial, and Michael E. Saks. An optimal on-line algorithm for metrical task system. *J. ACM*, 39(4):745–763, 1992. doi:10.1145/146585.146588.

9 Peter Braun and Heiner Litz. Understanding memory access patterns for prefetching. In *International Workshop on AI-assisted Design for Architecture (AIDArc), held in conjunction with ISCA*, 2019. URL: https://eecs.oregonstate.edu/aidarc/paper/MAP.pdf.

10 Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, USA, 2006. doi:10.1017/CBO9780511546921.

11 Yuval Emek, Shay Kutten, and Yangguang Shi. Online paging with a vanishing regret. *arXiv preprint*, 2020. arXiv:2011.09439.

12 Yuval Emek, Ron Lavi, Rad Niazadeh, and Yangguang Shi. Stateful posted pricing with vanishing regret via dynamic deterministic markov decision processes. *Advances in Neural Information Processing Systems (NeurIPS '20)*, 33, 2020. URL: https://papers.nips.cc/paper/2020/file/1f10c3650a3aa5912dccc5789fd515e8-Paper.pdf.

13 Amos Fiat, Richard M. Karp, Michael Luby, Lyle A. McGeoch, Daniel Dominic Sleator, and Neal E. Young. Competitive paging algorithms. *Journal of Algorithms*, 12(4):685–699, 1991. doi:10.1016/0196-6774(91)90041-V.

14 Sreenivas Gollapudi and Debmalya Panigrahi. Online algorithms for rent-or-buy with expert advice. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *ICML '19*, pages 2319–2327, Long Beach, California, USA, June 2019. PMLR. URL: http://proceedings.mlr.press/v97/gollapudi19a.html.

15 Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations*, ICLR '15, San Diego, CA, USA, May 2015. arXiv:1412.6572.

16 Milad Hashemi, Kevin Swersky, Jamie A. Smith, Grant Ayers, Heiner Litz, Jichuan Chang, Christos Kozyrakis, and Parthasarathy Ranganathan. Learning memory access patterns. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *ICML '18*, pages 1924–1933, Stockholmsmässan, Stockholm, Sweden, July 2018. PMLR. URL: http://proceedings.mlr.press/v80/hashemi18a.html.

17 Evan Zheran Liu, Milad Hashemi, Kevin Swersky, Parthasarathy Ranganathan, and Junwhan Ahn. An imitation learning approach for cache replacement. In *Proceedings of the 37th International Conference on Machine Learning*, ICML '20, July 2020. arXiv:2006.16239.

18 Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. In *Proceedings of the 35th International Conference on Machine Learning, ICML '18*, volume 80, pages 3302–3311, Stockholmsmässan, Stockholm, Sweden, July 2018. PMLR. URL: http://proceedings.mlr.press/v80/lykouris18a.html.

19 Leeor Peled, Uri C. Weiser, and Yoav Etsion. A neural network prefetcher for arbitrary memory access patterns. *ACM Transactions on Architecture and Code Optimization (TACO)*, 16(4):37:1–37:27, 2020. doi:10.1145/3345000.

20 Dhruv Rohatgi. Near-optimal bounds for online caching with machine learned advice. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA '20*, pages 1834–1845, Salt Lake City, UT, USA, January 2020. SIAM. doi:10.1137/1.9781611975994.112.

21 Daniel D. Sleator and Robert E. Tarjan. Amortized efficiency of list update and paging rules. *Commun. ACM*, 28(2):202–208, 1985. doi:10.1145/2786.2793.

22 Ajitesh Srivastava, Angelos Lazaris, Benjamin Brooks, Rajgopal Kannan, and Viktor K. Prasanna. Predicting memory accesses: the road to compact ml-driven prefetcher. In *Proceedings of the International Symposium on Memory Systems*, MEMSYS '19, pages 461–470, Washington, DC, USA, September – October 2019. ACM. doi:10.1145/3357526.3357549.

23 Ajitesh Srivastava, Ta-Yang Wang, Pengmiao Zhang, César Augusto Fonticielha De Rose, Rajgopal Kannan, and Viktor K. Prasanna. Memmap: Compact and generalizable meta-lstm models for memory access prediction. In *Advances in Knowledge Discovery and Data Mining - 24th Pacific-Asia Conference*, volume 12085 of *PAKDD '20*, pages 57–68, Singapore, May 2020. Springer. doi:10.1007/978-3-030-47436-2_5.

24 Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations*, ICLR '14, Banff, AB, Canada, April 2014. arXiv:1312.6199.

25 Alexander Wei. Better and simpler learning-augmented online caching. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, volume 176 of *APPROX/RANDOM '20*, pages 60:1–60:17, Virtual Conference, August 2020. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.APPROX/RANDOM.2020.60.

26 William A. Wulf and Sally A. McKee. Hitting the memory wall: implications of the obvious. *SIGARCH Computer Architecture News*, 23(1):20–24, 1995. doi:10.1145/216585.216588.