Optimal replacement policies for non-uniform cache objects with optional eviction

Omri Bahat   Armand M. Makowski
Department of Electrical and Computer Engineering
and the Institute for Systems Research
University of Maryland College Park
College Park, Maryland 20742
Email: Obahat@glue.umd.edu armand@isr.umd.edu

Abstract— Replacement policies for general caching applications and Web caching in particular have been discussed extensively in the literature. Many ad-hoc policies have been proposed that attempt to take adavantage of the retrieval latency of documents, their size, the popularity of references and temporal locality of requested documents. However, the problem of finding optimal replacement policies under these factors has not been pursued in any systematic manner. In this paper, we take a step in that direction: We first show, still under the Independent Reference Model, that a simple Markov stationary replacement policy, called the policy $C_0^\star$, minimizes the long-run average metric induced by non-uniform document costs when document eviction is optional. We then propose a framework for operating caching systems with multiple performance metrics. We do so by solving a constrained caching problem with a single constraint. The resulting constrained optimal replacement policy is obtained by simple randomization between two Markov stationary optimal replacement policies $C_0^\star$ but induced by different costs.

I. INTRODUCTION

Web caching aims to reduce network traffic, server load and user-perceived retrieval latency by replicating "popular" content on proxy caches that are strategically placed within the network. Key to the effectiveness of such proxy caches is the implementation of document replacement algorithms that can yield high hit ratios. A large number of techniques for file caching and virtual memory replacement have been developed [1], [2], [5] but unfortunately they do not necessarily transfer to Web caching as explained below. Despite the ever decreasing price of storage devices, the optimization or fine tuning of cache replacement policies is not a moot point for the benefits of even slight improvements in cache performance can have an appreciable effect on network traffic, especially when such gains are compounded through a hierarchy of caches.

In the context of conventional caching the underlying working assumption is the so-called Independent Reference Model, whereby document requests are assumed to form an i.i.d. sequence. It has been known for some time [1], [2], [5] that the miss rate (equivalently, the hit rate) is minimized (equivalently, maximized) by the so-called policy A_0 according to which a document is evicted from the cache if it has the smallest probability of occurence (equivalently, is the least popular) among the documents in the cache. More precisely, let $doc(1), \ldots, doc(N)$ denote the set of documents to be requested and let $p(j)$ denote the probability of reference

of $doc(j)$ $(j = 1, \ldots, N)$. When the cache is full and the requested document is not in the cache, A_0 prescribes

$$\text{Evict } doc(i)$$
$$\text{if } \quad i = \arg\min\left(p(j) : \ doc(j) \text{ in cache}\right). \quad (1)$$

In practice, the popularity vector $\mathbf{p} = (p(1), \ldots, p(N))$ is not available and needs to be estimated on-line from incoming requests. This naturally gives rise to the LFU (Least Frequently Used) policy which mimics A_0 through the Certainty Equivalence Principle: When the cache is full and the kth requested document is not in the cache, LFU prescribes

$$\text{Evict } doc(i)$$
$$\text{if } \quad i = \arg\min\left(\hat{p}_k(j) : \ doc(j) \text{ in cache}\right) \quad (2)$$

where $\hat{p}_k(j)$ is the frequency estimate of $p(j)$ based on the trace measurements up to the kth request. The focus on miss and hit rates as performance criteria is reflective of the fact that historically, pages in memory systems were of equal size, and transfer times of pages from the primary storage to the cache were nearly constant over time and independent of the document transferred.

Interestingly enough, even in this restricted context, the popularity information as derived from the relative access frequencies of objects requested through the cache, is seldom maintained and is rarely used directly in the design of cache replacement policies. This is so because of the difficulty to capture this information in an on-line fashion in contrast with other attributes of the request stream, said attributes being thought indicative of the future popularity of requested objects. Typical examples include temporal locality via the recency of access and object size which lead very naturally to the Least-Recently-Used (LRU) and Largest-File-First (LFF) replacement policies, respectively.

At this point it is worth stressing the three primary differences between Web caching and conventional caching:

(1): Web objects or documents are of variable size whereas conventional caching handles fixed-size documents or pages. Neither the policy A_0 nor the LRU policy (nor many other policies proposed in the literature on conventional caching) account for the variable size of documents;

(2): The miss penalty or retrieval cost of missed documents from the server to the proxy can vary significantly over time and per document. In fact, the cost value may not be known in advance and must sometimes be estimated on-line before a decision is taken. For instance, the download time of a Web page depends on the size of the document to be retrieved, on the available bandwidth from the server to the cache, and on the route used. These factors may vary over time due to changing network conditions (e.g., link failure or network overload);

(3) Access streams seen by the proxy cache are the union of Web access streams from tens to thousands of users, instead of coming from a few programmed sources as is the case in virtual memory paging, so the Independent Reference Model is not likely to provide a good fit to Web traces. In fact, Web traffic patterns were found to exhibit temporal locality (i.e., temporal correlations) in that recently accessed objects are more likely to be accessed in the near future. To complicate matters, the popularity of Web objects was found to be highly variable (i.e., bursty) over short time scales but much smoother over long time scales.

These differences, namely variable size, variable cost and the more complex statistics of request patterns, preclude an easy transfer of caching techniques developed earlier for computer system memory. A large number of studies have focused on the design of efficient replacement policies, e.g., see [8] [9] [10] [11] and references therein for a sample literature. Proposed policies typically exploit either access recency (e.g., the LRU policy) or access frequency (e.g., the LFU policy) or a combination thereof (e.g., the hybrid LRFU policy). The numerous policies which have been proposed are often ad-hoc attempts to take advantage of the statistical information contained in the stream of requests, and to address the factors (1)-(3) above. Their performance is typically evaluated via trace-driven simulations, and compared to that of other well-established policies.

As should be clear from the discussion above, the classical set-up used in [1], [2] and [5] is too restrictive to capture the salient features present in Web caching. Indeed, the Independent Reference Model fails to simultaneously capture both popularity (i.e., long-term frequencies of requested documents) and temporal locality (i.e., correlations among document requests). It also does not account for documents with variable sizes. Moreover, this literature implicitly assumes that document replacement is *mandatory* upon a cache miss, i.e., a requested document not found in cache must be put in the cache. While this requirement is understandable when managing computer memory, it is not as crucial when considering Web caches,[1] especially if this approach results in simple document replacement policies with good performance.

With these difficulties in mind, it seems natural to seek to extend provably optimal caching policies in several directions: (i) The documents have non-uniform costs as we assimilate

size and variable retrieval latency to cost, with $c(j)$ denoting the cost of retrieving $doc(j)$ $(j = 1, \ldots, N)$, (ii) there exist correlations in the request streams, and (iii) document (re)placement is optional upon a cache miss.

In this paper, we take an initial step in the directions (i) and (iii): While still retaining the Independent Reference Model, we consider the problem of finding an optimal replacement policy with *non*-uniform retrieval cost $(c(1), \ldots, c(N))$ under the option that a requested document not in cache is not necessarily put in cache after being retrieved from the server. Interestingly enough, this simple change in operational constraints allows us to determine completely the structure of the replacement policy that minimizes the average cost criterion (over both finite and infinite horizons). Making use of standard ideas from the theory of Markov Decision Processes (MDPs) [6], [13], we show [Theorem 1] that the optimal policy is the (non-randomized) Markov stationary policy $C_0^\star$ that prescribes

$$\text{Evict } doc(i) \quad \text{if} \tag{3}$$

$$i = \arg\min(p(j)c(j) : doc(j) \text{ in cache or request}).$$

The simplicity of this optimal replacement policy should be contrasted with the state of affairs in the traditional formulation when replacement is mandatory. Indeed, except for the optimality of A_0 for uniform cache objects, i.e., $c(1) = \ldots = c(N)$, there are no known results concerning the structure of the optimal policy for an *arbitrary* cost structure (to the best of the authors' knowledge). It is tempting, yet erroneous, to conclude that the simple stationary Markov replacement policy C_0 that prescribes

$$\text{Evict } doc(i) \quad \text{if} \tag{4}$$

$$i = \arg\min(p(j)c(j) : doc(j) \text{ in cache})$$

is optimal; this policy is "myopically" optimal but usually not optimal as simple examples show. Curiously, the policy (4) is reminiscent of, and similar to, the policy $C_0^\star$ given in (3).

The ability to find provably optimal policies under an arbitrary cost structure can be put to advantage as follows: As in most complex engineering systems, multiple performance metrics need to be considered when operating caches, sometimes leading to conflicting objectives. For instance, managing the cache to achieve as small a miss rate as possible does not necessarily ensure that the average latency of retrieved documents is as small as could be since the latter performance metric typically depends on the size of retrieved documents while the former does not. One possible approach to capture multi-criteria aspects is to introduce constraints. In the second part of the paper, still under optional eviction, we formulate the problem of finding a replacement policy that minimizes an average cost under a single constraint in terms of another long-run average metric. We identify the structure of the constrained optimal policy as a randomized Markov stationary policy obtained by randomizing two simple policies of the type (3). The analysis relies on a simplified version of a methodology developed in the context of MDPs with a constraint in [3].

---

[1] In Web caching timescales are slower than in conventional caching due to variable network latencies.

The paper is organized as follows: The search for optimal replacement policies with optional eviction is formulated as an MDP in Section II. Its solution is discussed in Section III and relationships between C_0 and $C_0^\star$ are explored in Section IV. Section V is devoted to the constrained problem. The proof of the key Theorem 1 is given in Appendix A.

II. FINDING GOOD REPLACEMENT POLICIES

One approach for designing good replacement policies is to couch the problem as one of sequential decision making under uncertainty. The analysis that establishes the optimality of the policy A_0 under the Independent Reference Model with mandatory eviction and uniform costs, is based on Dynamic Programming arguments as developed for MDPs [6] [13]. Here, we modify the MDP framework used in [1], [2], [5] in order to incorporate the possibility of optional eviction.

A. An MDP framework for caching under optional eviction

The system comprises a server where a copy of each of its N documents is available, and a cache of size M with $1 \leq M < N$. Documents are first requested at the cache: If the requested document has a copy already in cache (i.e., a hit), this copy is downloaded by the user at some cost (e.g., latency). If the requested document is not in cache (i.e., a miss), a copy is requested from the server to be put in the cache. If the cache is already full, then a decision needs to be taken as to whether a document already in cache will be evicted (to make place for the copy of document just requested) and if so, which one. This decision is taken on the basis of earlier decisions and past requests in order to minimize a cost function associated with the operation of the cache over an infinite horizon.

Decision epochs are defined as the instants at which requests for documents are presented at the cache, and are indexed by $t = 0, 1, \ldots$. At time t, let S_t denote the state of the cache, thus S_t is a subset of $\{1, \ldots, N\}$ with size $|S_t| \leq M$[2]. We introduce the $\{1, \ldots, N\}$-valued random variable (rv) R_t to encode the identity of the document requested at time t.

When the request R_t is made, the state of the cache is S_t and let U_t denote the action prompted by the request R_t. If the request R_t is already in cache, then we use the convention $U_t = 0$ to denote the fact that no replacement decision needs to be taken. On the other hand, if the request R_t is not in the cache, then U_t takes value in $S_t + R_t$ and identifies the document to be removed: If U_t is selected in S_t, then an eviction takes place with the document U_t removed from the cache and replaced by R_t. On the other hand if $U_t = R_t$, then no document is replaced. Thus, the resulting cache state S_{t+1}

(just before the next request R_{t+1} is made) is given by [3]

$$
\begin{aligned}
S_{t+1} &= T(S_t, R_t, U_t) \\
&= \begin{cases}
S_t & \text{if } R_t \in S_t \\
S_t + R_t & \text{if } R_t \notin S_t, |S_t| < M \\
S_t + R_t - U_t & \text{if } R_t \notin S_t, |S_t| = M.
\end{cases}
\end{aligned} \quad (5)
$$

In this formulation, eviction is *not* mandatory, i.e., a document is not necessarily evicted from the cache if the requested document is not in cache and the cache is full. This is reflected by the possible selection $U_t = R_t$ in (5) when R_t is *not* S_t and $|S_t| = M$.

The state variable at time $t = 0, 1, \ldots$ being the pair (S_t, R_t), we identify the state space of the MDP to be the set $\mathcal{X}$ given by $\mathcal{X} := \mathcal{S}_M \times \{1, \ldots, N\}$ where $\mathcal{S}_M$ denotes the collection of all subsets of $\{1, \ldots, N\}$ of size less or equal to M. However, under the assumed rules of operation, the cache will eventually become full at some time and will remain so from that time onward, i.e., given any initial cache S_0, there exists $\tau = \tau(S_0)$ finite such that $|S_{\tau+t}| = M$ for all $t = 0, 1, \ldots$. As we are concerned primarily with the average cost criterion, there is no loss of generality (as we do from now on) in assuming the space state to be $\mathcal{X}^\star$ (instead of the original $\mathcal{X}$) with

$$
\mathcal{X}^\star := \{(S, r) \in \mathcal{X} : |S| = M\}.
$$

The information available to make a decision U_t when the document R_t ($t = 0, 1, \ldots$) is requested, is encapsulated in the rv H_t defined recursively by

$$
H_{t+1} = (H_t, U_t, S_{t+1}, R_{t+1}), \quad t = 0, 1, \ldots
$$

with $H_0 = (S_0, R_0)$. Thus, the range $\mathcal{H}_t$ of H_t can be defined recursively by

$$
\mathcal{H}_{t+1} = \mathcal{H}_t \times \{0, \ldots, N\} \times \mathcal{X}^\star, \quad t = 0, 1, \ldots
$$

with $\mathcal{H}_0 = \mathcal{X}^\star$. The decision U_t implemented in response to request R_t is then given by

$$
U_t = \pi_t(H_t)
$$

for some mapping $\pi_t : \mathcal{H}_t \to \{0, 1, \ldots, N\}$ such that

$$
\pi_t(H_t) = 0, \quad R_t \in S_t \quad (6)
$$

and

$$
\pi_t(H_t) \in S_t + R_t, \quad R_t \notin S_t \quad (7)
$$

for all $t = 0, 1, \ldots$. Such a collection $\pi = (\pi_t, \ t = 0, 1, \ldots)$ defines the replacement (or eviction) policy π.

We shall find it useful to consider *randomized* policies which are now defined: A randomized replacement policy π is a collection $(\pi_t, \ t = 0, 1, \ldots)$ of mappings $\pi_t : \{0, 1, \ldots, N\} \times \mathcal{H}_t \to [0, 1]$ such that for all $t = 0, 1, \ldots$, we have

$$
\sum_{u=0}^{N} \pi_t(u; H_t) = 1 \quad (8)
$$

---

[2]Here and in what follows, $|S_t|$ denotes the cardinality of S_t.

[3]Throughout, for any subset S of $\{1, \ldots, N\}$ and any elements x and u in $\{1, \ldots, N\}$, we write $S + x - u$ to denote the subset of $\{1, \ldots, N\}$ obtained from S by adding x to it and removing u from the resulting set, in that order.

with

$$\pi_t(u; H_t) = \delta(u; 0), \quad R_t \in S_t \tag{9}$$

and

$$\pi_t(u; H_t) = 0, \quad R_t \notin S_t \text{ and } u \notin S_t + R_t \tag{10}$$

for all $u = 0, 1, \ldots, N$. The class of all (possibly randomized) replacement policies is denoted by $\mathcal{P}$. Obviously, (6) and (7) are compatible with (9) and (10), respectively.

If the non-randomized replacement policy π has the property that

$$U_t = f_t(S_t, R_t), \quad t = 0, 1, \ldots$$

for mappings $f_t : \mathcal{X}^\star \to \{0, 1, \ldots, N\}$. we say that π is a Markov policy. If in addition, $f_t = f$ for all $t = 0, 1, \ldots$ the policy is said to be a (non-randomized Markov) stationary policy, in which case the policy is identified with the mapping f itself. Similar definitions can be given for randomized Markov stationary policies [6].

Under the Independent Reference Model, the sequence of requests is a sequence $\{R_t, t = 0, 1, \ldots\}$ of i.i.d. $\{1, \ldots, N\}$-valued rv distributed according to some pmf $\mathbf{p} = (p(1), \ldots, p(N))$ on $\{1, \ldots, N\}$ with $p(i) > 0$ for all $i = 1, \ldots, N$. Let $\mathbf{P}$ denote the probability measure under the pmf $\mathbf{p}$, with corresponding expectation operator $\mathbf{E}$.

The definition of the underlying MDP is completed by associating with each admissible policy π in $\mathcal{P}$, a probability measure $\mathbf{P}_\pi$ defined through the following requirements[4]: For each $t = 0, 1, \ldots$, we have

$$\mathbf{P}_\pi[U_t = u|H_t] = \pi_t(u; H_t),^5 \quad u = 0, \ldots, N \tag{11}$$

and

$$
\begin{aligned}
&\mathbf{P}_\pi[S_{t+1} = S', R_{t+1} = y|H_t, U_t] \\
&= p(y)\mathbf{P}_\pi[S_{t+1} = S'|H_t, U_t] \\
&= p(y)\mathbf{1}[T(S_t, R_t, U_t) = S'] \tag{12}
\end{aligned}
$$

for every state (S', y) in $\mathcal{X}^\star$. Let $\mathbf{E}_\pi$ denote the expectation operator associated with the probability measure $\mathbf{P}_\pi$.

B. The cost functionals

With any one-step cost function $c : \{1, \ldots, N\} \to \mathbf{R}_+$, we associate several cost functions: Fix a replacement policy π in $\mathcal{P}$. For each $T = 0, 1, \ldots$, define the total cost over the horizon $[0, T]$ under the policy π by

$$J_c(\pi; T) = \mathbf{E}_\pi \left[\sum_{t=0}^{T} \mathbf{1}[R_t \notin S_t] c(R_t) \right].$$

The average cost (over the entire horizon) under the policy π is then defined by

$$
\begin{aligned}
J_c(\pi) &= \limsup_{T \to \infty} \frac{1}{T+1} J_c(\pi; T) \tag{13} \\
&= \limsup_{T \to \infty} \frac{1}{T+1} \mathbf{E}_\pi \left[\sum_{t=0}^{T} \mathbf{1}[R_t \notin S_t] c(R_t) \right].
\end{aligned}
$$

[4]This measure $\mathbf{P}_\pi$ is defined on the product space $(\mathcal{X}^\star \times \{0, 1, \ldots, N\})^\infty$ equipped with its natural Borel σ-field.

[5]This relation holds for randomized policies. For a non-randomized policy π, it takes the form $\mathbf{P}_\pi[U_t = u|H_t] = \delta(u, \pi_t(H_t))$ for all $u = 0, \ldots, N$.

We use the limsup operation in the definition above since under an arbitrary policy π the limit in (13) may not exist; this is standard practice in the theory of MDPs.

The basic problem we address is that of finding a cache replacement policy $\pi^\star$ in $\mathcal{P}$ such that

$$J_c(\pi^\star) \leq J_c(\pi), \quad \pi \in \mathcal{P}.$$

We refer to any such policy $\pi^\star$ as an optimal replacement policy (under the long-term average criterion); it is not necessarily unique. However, the state space $\mathcal{X}^\star$ and the action space $\{0, 1, \ldots, N\}$ being finite, it is well known that the optimal replacement policy $\pi^\star$ can always be selected to be a non-randomized Markov stationary policy [6][13, Chap. V]. In the process of identifying such an optimal policy $\pi^\star$ in the next section, we will need a notion of optimality for the finite horizon problems. More specifically, for each $T = 0, 1, \ldots$, a policy $\pi^\star$ in $\mathcal{P}$ is said to be an optimal replacement policy on the horizon $[0, T]$ if

$$J_c(\pi^\star; T) \leq J_c(\pi; T), \quad \pi \in \mathcal{P}.$$

Obviously, if the policy $\pi^\star$ in $\mathcal{P}$ is an optimal replacement policy on the horizon $[0, T]$ for *each* $T = 0, 1, \ldots$, then it is an optimal replacement policy under the long-term average criterion.

C. Examples

A number of situations can be handled by adequately specializing the cost-per-step c: Indeed, if $c(i) = 1$ ($i = 1, \ldots, N$), then $J_c(\pi; T)$ and $J_c(\pi)$ are the expected number of cache misses over the horizon $[0, T]$ and the average miss rate under policy π, respectively. On the other hand, if c is taken to be the size function $s : \{1, \ldots, N\} \to \mathbf{N}$, with $s(i)$ denoting the size (in bytes) of $doc(i)$ ($i = 1, \ldots, N$), then the byte hit rate under policy π can be defined by

$$BHR(\pi) = \liminf_{T \to \infty} \frac{\mathbf{E}_\pi \left[\sum_{t=0}^{T} \mathbf{1}[R_t \in S_t] s(R_t) \right]}{\mathbf{E}_\pi \left[\sum_{t=0}^{T} s(R_t) \right]} \tag{14}$$

where the liminf operation reflects the fact that this performance is maximized. To make use of the MDP framework used here, we first note that

$$\mathbf{E}_\pi \left[\sum_{t=0}^{T} s(R_t) \right] = (T+1)\mathbf{E}[s(R)]$$

for some $\{1, \ldots, N\}$-valued rv R with pmf $\mathbf{p}$. Next, we see that

$$
\begin{aligned}
&BHR(\pi) \\
&= 1 - \limsup_{T \to \infty} \frac{\mathbf{E}_\pi \left[\sum_{t=0}^{T} \mathbf{1}[R_t \notin S_t] s(R_t) \right]}{\mathbf{E}_\pi \left[\sum_{t=0}^{T} s(R_t) \right]} \\
&= 1 - \frac{J_s(\pi)}{\mathbf{E}[s(R)]}.
\end{aligned}
$$

Hence, maximizing the byte hit rate is equivalent to minimizing the average cost associated with s.

III. NON-UNIFORM COST OPTIMAL REPLACEMENT POLICY WITHOUT MANDATORY EVICTION

In this section we discuss the optimal cache replacement policy for non uniform costs under the Independent Reference Model when eviction is *not* mandatory. A useful characterization of this optimal policy is provided through the Dynamic Programming Equation (DPE) for the corresponding MDP [6], [13].

A. The optimal replacement policy

For each $T = 0, 1, \ldots$, we define the cost-to-go associated with the policy π in $\mathcal{P}$ starting in the initial state (S, r) in $\mathcal{X}^\star$ to be

$$J_c^\pi((S, r); T)$$
$$:= \mathbf{E}_\pi \left[\sum_{t=0}^{T} \mathbf{1}\left[R_t \notin S_t\right] c(R_t) | S_0 = S, R_0 = r \right].$$

Next, the value function over the horizon $[0, T]$ is defined by

$$V_T(S, r) := \inf_{\pi \in \mathcal{P}} J_c^\pi((S, r); T), \quad (S, r) \in \mathcal{X}^\star.$$

For the MDP at hand, the DPE takes the form

$$
\begin{aligned}
&V_{T+1}(S, r) \\
&= \mathbf{1}\left[r \in S\right] \mathbf{E}\left[V_T(S, R^*)\right] \\
&+ \mathbf{1}\left[r \notin S\right] \left(c(r) + \min_{u \in S+r} \mathbf{E}\left[V_T(S + r - u, R^*)\right] \right)
\end{aligned}
\tag{15}
$$

for every state (S, r) in $\mathcal{X}^\star$ with R^* denoting an $\{1, \ldots, N\}$-valued rv with pmf $\mathbf{p}$. The possibility of non-eviction is reflected in the choice $u = r$ (obviously in $S + r$). Moreover, as well known [13], the optimal action to be taken in state (S, r) at time $t = 0$ when minimizing the cost criterion over the horizon $[0, T]$ is simply given by

$$g_T^\star(S, r) := \arg \min_{u \in S+r} \left(\mathbf{E}\left[V_T(S + r - u, R^*)\right] \right),$$

with a lexicographic tie-braker for sake of concreteness. We set $g_T^\star(S, r) = 0$ whenever r is in S. The next result presents a complete characterization of $g_T^\star : \mathcal{X}^\star \to \{0, 1, \ldots, N\}$.

Theorem 1: For each $T = 0, 1, \ldots$, we have the identification

$$g_T^\star(S, r) = g_\star(S, r) \tag{16}$$

for any state (S, r) in $\mathcal{X}^\star$ whenever r is not in S, with

$$g_\star(S, r) := \arg \min_{u \in S+r} \left(p(u)c(u) \right). \tag{17}$$

The proof of Theorem 1 is given in Appendix A. Note that $g_T^\star$ does not depend on T, and that the non-randomized Markov stationary policy associated with $g_\star$ is the policy $C_0^\star$ introduced earlier. It is now plain from Theorem 1 that the Markov stationary policy $C_0^\star$ is optimal for both the finite and infinite horizon cost problems.

B. Evaluation of the optimal cost

In order to calculate the average cost, byte hit rate, and other interesting properties of the replacement policy of Theorem 1, we find it useful to introduce the permutation σ of $\{1, \ldots, N\}$ which orders the values $p(i)c(i)$ $(i = 1, \ldots, N)$ in decreasing order, namely

$$p(\sigma(1))c(\sigma(1)) \geq p(\sigma(2))c(\sigma(2)) \geq \ldots \tag{18}$$

The key observation is that the long term usage of the optimal replacement policy $C_0^\star$ results in a set of M fixed documents in the cache, namely $\{\sigma(1), \ldots, \sigma(M)\}$, so that every document in the set $\{\sigma(1), \ldots, \sigma(M)\}$ is never evicted from the cache once requested. If we write

$$S := \{\sigma(1), \ldots, \sigma(M)\} \tag{19}$$

for this steady-state stack, then formally

$$\lim_{t \to \infty} \mathbf{P}_{C_0^\star}\left[\sigma(i) \in S_t\right] = 0, \quad i = M + 1, \ldots, N$$

and

$$J_d(C_0^\star) = \sum_{i \notin S} p(i)d(i) = \sum_{i=M+1}^{N} p(\sigma(i))d(\sigma(i)) \tag{20}$$

for any cost $d : \{1, \ldots, N\} \to \mathbb{R}_+$ (and in particular the cost $c : \{1, \ldots, N\} \to \mathbb{R}_+$ which induces the policy $C_0^\star$). Thus, the byte hit rate associated with the policy $C_0^\star$ is simply given by

$$BHR(C_0^\star) = \frac{\sum_{j=1}^{M} p(\sigma(j))s(\sigma(j))}{\mathbf{E}\left[s(R)\right]}. \tag{21}$$

Another interesting observation is the relation of the optimal replacement policy $C_0^\star$ to the well-established Greedy Dual* and Greedy Dual-Size replacement policies described in [8] and [9]. Let $c_{GD} : \{1, \ldots, N\} \to \mathbb{R}_+$ be an arbitrary cost used by the Greedy Dual policies. Under optional eviction, the Greedy Dual policies in case of a cache miss prescribe

$$\text{Evict } doc(i) \tag{22}$$
$$\text{if} \quad i = \arg \min_{j \in S+r} \left(L + \left(\frac{p(j)c_{GD}(j)}{s(j)} \right)^{\frac{1}{\beta}} \right)$$

where L is a contribution of the temporal locality of reference to the replacement policy and $\beta > 0$ is a weight factor that modulates the contribution of the probability of reference, document size and document cost to the eviction decision. Under the Independent Reference Model, we can take $L = 0$, in which case the Greedy Dual policies simplify to

$$\text{Evict } doc(i) \quad \text{if} \quad i = \arg \min_{j \in S+r} \left(\frac{p(j)c_{GD}(j)}{s(j)} \right).$$

This is a special case of the optimal replacement policy $C_0^\star$ associated with cost function $c : \{1, \ldots, N\} \to \mathbb{R}_+$ given by

$$c(i) := \frac{c_{GD}(i)}{s(i)}, \quad i = 1, \ldots, N.$$

C. Implementing the optimal policy

A natural implementation of the optimal replacement policy $C_0^\star$ is achieved by invoking the Certainty Equivalence Principle. In addition to the online estimation of the probability of references (as was the case for (2)), this approach now requires the estimation of additional parameters which enter the definition of the overall document cost $(c(j), \quad j = 1, \ldots, N)$, e.g., in the case of document latency, the document size might be fully known but the available bandwidth to the server needs to be measured online at request time. Let $(\hat{c}_k(j), \ j = 1, \ldots, N)$ denote estimates of the document costs which are available at the cache at the time instance of the kth request: If $|S_t| < M$, document placement always takes place; otherwise the replacement action is dictated by

$$\text{Evict } doc(i) \quad \text{if} \quad i = \arg \min_{j \in S_t + R_t} (\hat{p}_k(j)\hat{c}_k(j)).$$

IV. (NON)-OPTIMALITY OF C_0

Mandatory eviction can be incorporated into the MDP framework of Section II by strengthening (7) and (10) to read

$$\pi_t(H_t) \in S_t, \quad R_t \notin S_t \tag{23}$$

for non-randomized policies, and

$$\pi(u; H_t) = 0, \quad R_t \notin S_t \text{ and } u \notin S_t \tag{24}$$

for randomized policies for all $t = 0, 1, \ldots$, respectively. Let $\mathcal{P}_{\text{Mand}}$ denote the class of all (possibly randomized) replacement policies in $\mathcal{P}$ which enforce mandatory eviction. The set of policies $\mathcal{P}_{\text{Mand}}$ being a subset of $\mathcal{P}$, it is plain that

$$\inf_{\pi \in \mathcal{P}} J_c(\pi) \leq \inf_{\pi \in \mathcal{P}_{\text{Mand}}} J_c(\pi). \tag{25}$$

It is well known [1], [2], [5] that for any cost function $c : \{1, \ldots, N\} \to \mathbb{R}_+$, the cost associated with the policy C_0 induced by c (via (4)) is given by

$$\begin{aligned} J_c(C_0) &= \sum_{i=M}^{N} p(\sigma(i))c(\sigma(i)) \\ &\quad - \frac{\sum_{i=M}^{N} p(\sigma(i))^2 c(\sigma(i))}{\sum_{i=M}^{N} p(\sigma(i))} \end{aligned} \tag{26}$$

under the convention (18). Using (20) (with $d = c$), we see that

$$J_c(C_0) - J_c(C_0^\star)$$
$$= p(\sigma(M))c(\sigma(M)) - \frac{\sum_{i=M}^{N} p(\sigma(i))^2 c(\sigma(i))}{\sum_{i=M}^{N} p(\sigma(i))}$$
$$= \frac{\sum_{i=M}^{N} p(\sigma(i)) \left(p(\sigma(M))c(\sigma(M)) - p(\sigma(i))c(\sigma(i))\right)}{\sum_{i=M}^{N} p(\sigma(i))}$$

so that $J_c(C_0^\star) \leq J_c(C_0)$ in agreement with (25). Moreover, $J_c(C_0^\star) = J_c(C_0)$ if and only if

$$p(\sigma(i))c(\sigma(i)) = p(\sigma(M))c(\sigma(M)), \quad i = M, M+1, \ldots, N$$

as would be expected, in which case the policy C_0 is indeed optimal amongst all replacement policies enforcing mandatory eviction when the cache is full.

However, though very tempting in view of the structure of the policy A_0 (which is optimal in the uniform cost case), the policy C_0 will not be optimal in general as can be seen on simple examples.[6] Under non-uniform costs, the deterministic version of the page replacement problem is known to be NP-complete [7] (in contrast with the uniform case where Belady's algorithm is optimal). Policy C_0 was incorrectly claimed to be optimal in [14] where it was used to produce efficient randomized replacement policies.

V. OPTIMAL CACHING UNDER A CONSTRAINT

One possible approach to capture the multi-criteria aspect of running caching systems is to introduce constraints. Here, we revisit the caching problem studied in Section III under a single constraint.

A. Problem Formulation

Formulating the caching problem under a single constraint requires two cost functions, say $c, d : \{1, \ldots, N\} \to \mathbb{R}_+$. As before, $c(R_t)$ and $d(R_t)$ represent different costs of retrieving the requested document R_t if not in the cache S_t at time t. For instance, we could take

$$c(i) = 1 \quad \text{and} \quad d(i) = s(i), \ i = 1, \ldots, N \tag{27}$$

to reflect interest in miss rate and document retrieval latency, respectively.

The problem of interest can now be formulated as follows: Given some $\alpha > 0$, we say that the policy π in $\mathcal{P}$ satisfies the constraint at level α if

$$J_d(\pi) \leq \alpha. \tag{28}$$

Let $\mathcal{P}(d; \alpha)$ denote the class of all cache replacement policies in $\mathcal{P}$ that satisfy the constraint (28).

The problem is to find a cache replacement policy $\pi^\star$ in $\mathcal{P}(d; \alpha)$ such that

$$J_c(\pi^\star) \leq J_c(\pi), \quad \pi \in \mathcal{P}(d; \alpha).$$

We refer to any such policy $\pi^\star$ as a constrained optimal policy (at level α). With the choice (27) this formulation would focus on minimizing the miss rate with a bound on average latency of document retrieval (under the assumption that retrieval latency is proportional to the size of the document to be retrieved).

One natural approach to solving this problem is to consider the corresponding Lagrangian functional defined by

$$\tilde{J}_\lambda(\pi) = J_c(\pi) + \lambda J_d(\pi), \quad \pi \in \mathcal{P}, \ \lambda \geq 0. \tag{29}$$

The basic idea is then to find for each $\lambda \geq 0$, a cache replacement policy $\pi^\star(\lambda)$ in $\mathcal{P}$ such that

$$\tilde{J}_\lambda(\pi^\star(\lambda)) \leq \tilde{J}_\lambda(\pi), \quad \pi \in \mathcal{P}. \tag{30}$$

Now, if for some $\lambda^\star \geq 0$, the policy $\pi^\star(\lambda^\star)$ happens to *saturate* the constraint at level α, i.e., $J_d(\pi^\star(\lambda^\star)) = \alpha$, then the policy $\pi^\star(\lambda^\star)$ belongs to $\mathcal{P}(d; \alpha)$ and its optimality implies

$$\tilde{J}_{\lambda^\star}(\pi^\star(\lambda^\star)) \leq \tilde{J}_{\lambda^\star}(\pi), \quad \pi \in \mathcal{P}.$$

[6]Take $N = 3$, $M = 2$, $\mathbf{p} = (.009, .001, .99)$ and costs $(20, 5, 1)$.

In particular, for any policy π in $\mathcal{P}(d; \alpha)$, this last inequality readily leads to

$$J_c(\pi^\star(\lambda^\star)) \leq J_c(\pi), \quad \pi \in \mathcal{P}(d; \alpha),$$

and the policy $\pi^\star(\lambda^\star)$ solves the constrained optimization problem.

The only glitch in this approach resides in the use of the limsup operation in the definition (13), so that $\tilde{J}_\lambda(\pi)$ is not necessarily the long-run average cost under policy π for some appropriate one-step cost. Thus, finding the optimal cache replacement policy $\pi^\star(\lambda)$ specified by (30) cannot be achieved in a straightforward manner.

B. A Lagrangian approach

Following the treatment in [3], we now introduce an alternate Lagrangian formulation which circumvents this technical difficulty and allows us eventually to carry out the program outlined above: For each $\lambda \geq 0$, we define the one-step cost function $b_\lambda : \{1, \ldots, N\} \to \mathbb{R}_+$ by

$$b_\lambda(i) := c(i) + \lambda d(i), \quad i = 1, \ldots, N$$

and consider the corresponding long-run average functional (13), i.e., for any policy π in $\mathcal{P}$, we set

$$
\begin{aligned}
J_\lambda(\pi) &:= J_{b_\lambda}(\pi) \qquad\qquad\qquad (31) \\
&= \limsup_{T \to \infty} \frac{1}{T+1} \mathbf{E}_\pi \left[\sum_{t=0}^{T} \mathbf{1}\left[R_t \notin S_t \right] b_\lambda(R_t) \right].
\end{aligned}
$$

With these definitions we get

$$J_{b_\lambda}(\pi) \leq \tilde{J}_\lambda(\pi), \quad \pi \in \mathcal{P}$$

by standard properties of the limsup, with equality

$$J_{b_\lambda}(\pi) = \tilde{J}_\lambda(\pi)$$

whenever π is a Markov stationary policy.

For each $\lambda \geq 0$, the (unconstrained) caching problem associated with the cost b_λ is an MDP with finite state and action spaces. Thus, there exists a non-randomized Markov stationary policy, denoted g_λ, which is optimal [6], i.e.,

$$J_\lambda(g_\lambda) \leq J_\lambda(\pi), \quad \pi \in \mathcal{P}$$

and earlier remarks yield

$$\tilde{J}_\lambda(g_\lambda) \leq \tilde{J}_\lambda(\pi), \quad \pi \in \mathcal{P}.$$

In other words, the Markov stationary policy g_λ also minimizes the Lagrangian functional (29), and the relation

$$J_\lambda(g_\lambda) = \inf_{\pi \in \mathcal{P}} J_\lambda(\pi) = \inf_{\pi \in \mathcal{P}} \tilde{J}_\lambda(\pi) \qquad (32)$$

holds. Consequently, as argued in Section V-A, if for some $\lambda^\star \geq 0$, the policy $g_{\lambda^\star}$ saturates the constraint at level α, then the policy $g_{\lambda^\star}$ will solve the constrained optimization problem.

The difficulty of course is that *a priori* we may have $J_\lambda(g_\lambda) \neq \alpha$ for *all* $\lambda \geq 0$. However, the arguments given above still show that the search for the constrained optimal

policy can be recast as the problem of finding $\gamma \geq 0$ and a (possibly randomized) Markov stationary policy $g^\star$ such that

$$J_d(g^\star) = \alpha \qquad (33)$$

and

$$J_\gamma(g^\star) \leq J_\gamma(\pi), \quad \pi \in \mathcal{P}. \qquad (34)$$

C. On the way to solving the constrained MDP

The appropriate multiplier γ and the policy $g^\star$ appearing in (33) and (34) will be identified in Section V-D. To help us in this process we need some technical facts and notation which we now develop.

Theorem 2: The optimal cost function $\lambda \to J_\lambda(g_\lambda)$ is a non-decreasing concave function which is piecewise linear on $\mathbb{R}_+$.

Some observations are in order before giving a proof of Theorem 2: Fix $\lambda \geq 0$. In view of Theorem 1 we can select g_λ as the policy $C_0^\star$ induced by b_λ, i.e.,

$$\text{Evict } doc(i) \quad \text{iff} \quad i = \arg\min_{j \in S+r} (c(j) + \lambda d(j)). \qquad (35)$$

Let σ_λ denote the permutation of $\{1, \ldots, N\}$ which orders the values $p(i)b_\lambda(i)$ $(i = 1, \ldots, N)$ in decreasing order, namely

$$p(\sigma_\lambda(1))b_\lambda(\sigma_\lambda(1)) \geq p(\sigma_\lambda(2))b_\lambda(\sigma_\lambda(2)) \geq \ldots \qquad (36)$$

with a lexicographic tie-breaker. Let $S(\lambda)$ denote the steady-state stack induced by the policy g_λ, namely the collection of documents in the cache that results from long-term usage of the policy g_λ. Obviously, we have [7]

$$S(\lambda) = \{\sigma_\lambda(1), \ldots, \sigma_\lambda(M)\} \qquad (37)$$

so that

$$J_\lambda(g_\lambda) = J_{b_\lambda}(g_\lambda) = \sum_{i \notin S(\lambda)} p(i)b_\lambda(i) \qquad (38)$$

upon rephrasing comments made earlier in Section III.

Given the affine nature (in the variable λ) of the cost, there must exist a finite and strictly increasing sequence of non-zero scalar values $\lambda_1, \ldots, \lambda_L$ in $\mathbb{R}_+$ with $0 < \lambda_1 < \ldots < \lambda_L$ such that for each $\ell = 0, \ldots, L$, it holds that

$$S(\lambda) = S(\lambda_\ell), \quad \lambda \in I_\ell := [\lambda_\ell, \lambda_{\ell+1})$$

with the convention $\lambda_0 = 0$ and $\lambda_{L+1} = \infty$, but with

$$S(\lambda_\ell) \neq S(\lambda_{\ell+1}), \quad \ell = 0, \ldots, L-1.$$

In view of (38) it is plain that

$$J_\lambda(g_\lambda) = \sum_{i \notin S(\lambda_\ell)} p(i)b_\lambda(i) \qquad (39)$$

whenever λ belongs to I_ℓ for some $\ell = 0, \ldots, L$.

Proof. For each policy π in $\mathcal{P}$, the quantities $J_c(\pi)$ and $J_d(\pi)$ are non-negative as the one-step cost functions c and d are assumed non-negative. Thus, the mapping $\lambda \to \tilde{J}_\lambda(\pi)$

---

[7] The steady-state stack S given by (19) corresponds to the case $\lambda = 0$ with $\sigma_0 = \sigma$.

is non-decreasing and affine, and we conclude from (32) that the mapping $\lambda \to J_\lambda(g_\lambda)$ is indeed non-decreasing and concave. Its piecewise-linear character is a straightforward consequence of (39). ∎

In order to proceed we now make the following simplifying assumption.

(A) If for some $\lambda \geq 0$, it holds that

$$p(i)b_\lambda(i) = p(j)b_\lambda(j)$$

for some distinct $i, j = 1, \ldots, N$, then there does not exist any $k \neq i, j$ with $k = 1, \ldots, N$ such that

$$p(i)b_\lambda(i) = p(j)b_\lambda(j) = p(k)b_\lambda(k).$$

Assumption (A) can be removed at the cost of a more delicate analysis without affecting the essence of the optimality result to be derived shortly.

For each $\ell = 0, 1, \ldots, L$, the relative position of the quantities $p(i)b_\lambda(i)$ $(i = 1, \ldots, N)$ remains unchanged as λ sweeps through the interval $(\lambda_\ell, \lambda_{\ell+1})$. Under (A), when going through $\lambda = \lambda_{\ell+1}$, a *single* reversal occurs in the relative position with

$$S(\lambda_\ell) = \{\sigma_{\lambda_\ell}(1), \ldots, \sigma_{\lambda_\ell}(M-1), \sigma_{\lambda_\ell}(M)\}$$

and

$$S(\lambda_{\ell+1}) = \{\sigma_{\lambda_\ell}(1), \ldots, \sigma_{\lambda_\ell}(M-1), \sigma_{\lambda_\ell}(M+1)\}.$$

By continuity we must have

$$p(\sigma_{\lambda_\ell}(M))b_{\lambda_{\ell+1}}(\sigma_{\lambda_\ell}(M))$$
$$= p(\sigma_{\lambda_\ell}(M+1))b_{\lambda_{\ell+1}}(\sigma_{\lambda_\ell}(M+1)). \qquad (40)$$

Theorem 3: Under Assumption (A), the mapping $\lambda \to J_d(g_\lambda)$ is a non-increasing piecewise constant function on $\mathbb{R}_+$.

Proof. The analog of (39) holds in the form

$$J_d(g_\lambda) = \sum_{i \notin S(\lambda_\ell)} p(i)d(i) \qquad (41)$$

whenever λ belongs to I_ℓ for some $\ell = 0, \ldots, L$. Hence, the mapping $\lambda \to J_d(g_\lambda)$ is piecewise constant.

Now pick $\ell = 0, 1, \ldots, L-1$ and consider λ and μ in the *open* intervals $(\lambda_\ell, \lambda_{\ell+1})$ and $(\lambda_{\ell+1}, \lambda_{\ell+2})$, respectively. The desired monotonicty will be established if we can show that $J_d(g_\mu) - J_d(g_\lambda) \leq 0$. First, from (41), we note

$$J_d(g_\mu) - J_d(g_\lambda) \qquad (42)$$
$$= \sum_{i \in S(\lambda)} p(i)d(i) - \sum_{i \in S(\mu)} p(i)d(i)$$
$$= p(\sigma_{\lambda_\ell}(M))d(\sigma_{\lambda_\ell}(M))$$
$$\quad - p(\sigma_{\lambda_\ell}(M+1))d(\sigma_{\lambda_\ell}(M+1))$$

by comments made earlier as we recall that $S(\lambda) = S(\lambda_\ell)$ and $S(\mu) = S(\lambda_{\ell+1})$.

Next, pick $\varepsilon > 0$ such that $\lambda + \varepsilon$ and $\mu + \varepsilon$ are in the *open* intervals $(\lambda_\ell, \lambda_{\ell+1})$ and $(\lambda_{\ell+1}, \lambda_{\ell+2})$, respectively. By (39) we get $S(\lambda + \varepsilon) = S(\lambda)$ and

$$J_{\lambda+\varepsilon}(g_{\lambda+\varepsilon}) - J_\lambda(g_\lambda)$$
$$= \sum_{i \notin S(\lambda+\varepsilon)} p(i)b_{\lambda+\varepsilon}(i) - \sum_{i \notin S(\lambda)} p(i)b_\lambda(i)$$
$$= \sum_{i \notin S(\lambda)} p(i)b_{\lambda+\varepsilon}(i) - \sum_{i \notin S(\lambda)} p(i)b_\lambda(i)$$
$$= \varepsilon \sum_{i \notin S(\lambda)} p(i)d(i) \qquad (43)$$

Similarly,

$$J_{\mu+\varepsilon}(g_{\mu+\varepsilon}) - J_\mu(g_\mu) = \varepsilon \sum_{i \notin S(\mu)} p(i)d(i). \qquad (44)$$

By Theorem 2, the mapping $\lambda \to J_\lambda(g_\lambda)$ is concave, hence

$$J_{\mu+\varepsilon}(g_{\mu+\varepsilon}) - J_\mu(g_\mu) \leq J_{\lambda+\varepsilon}(g_{\lambda+\varepsilon}) - J_\lambda(g_\lambda).$$

Making use of (43) and (44) in this last inequality, we readily conclude that

$$\sum_{i \in S(\lambda)} p(i)d(i) \leq \sum_{i \in S(\mu)} p(i)d(i). \qquad (45)$$

But $S(\lambda) = S(\lambda_\ell)$ and $S(\mu) = S(\lambda_{\ell+1})$, whence (45) is equivalent to

$$p(\sigma_{\lambda_\ell}(M))d(\sigma_{\lambda_\ell}(M)) \leq p(\sigma_{\lambda_\ell}(M+1))d(\sigma_{\lambda_\ell}(M+1)).$$

The desired conclusion $J_d(g_\mu) - J_d(g_\lambda) \leq 0$ is now immediate from (42). ∎

D. The constrained optimal replacement policy

We are now ready to discuss the form of the optimal replacement policy for the constrained caching problem. Throughout we assume Assumption (A) to hold. Several cases need to be considered:

Case 1 – The unconstrained optimal replacement policy g_0 satisfies the constraint, i.e., $J_d(g_0) \leq \alpha$, in which case $g^\star$ is simply the optimal replacement policy $C_0^\star$ for the unconstrained caching problem. This case is trivial and requires no proof since by Theorem 1 the average cost is minimized and the constraint satisfied.

Case 2 – The unconstrained optimal replacement policy does not satisfy the constraint, i.e., $J_d(g_0) > \alpha$, but there exists $\lambda > 0$ such that $J_d(g_\lambda) \leq \alpha$. Two subcases of interest emerge and are presented in Theorems 4 and 5 below.

Case 2a – The situation when the policy g_λ above saturates the constraint at level α was covered earlier in the discussion; its proof is therefore omitted.

Theorem 4: If there exists $\lambda > 0$ such that $J_d(g_\lambda) = \alpha$, then the policy g_λ can be taken as the optimal replacement policy $g^\star$ for the constrained caching problem (and the constraint is saturated).

Case 2b – The case of greatest interest arises when the conditions of Theorem 4 are not met, i.e., $J_d(g_0) > \alpha$, $J_d(g_\mu) \neq \alpha$ for all $\mu \geq 0$ but there exists $\lambda > 0$ such that $J_d(g_\lambda) < \alpha$. In that case, by the monotonicity result of Theorem 3, the quantity

$$\gamma := \inf\{\lambda \geq 0 : J_d(g_\lambda) \leq \alpha\}$$

is a well defined scalar in $(0, \infty)$. In fact, we have the identification

$$\gamma = \lambda_{\ell+1} \tag{46}$$

for some $\ell = 0, 1, \ldots, L-1$, and it holds that

$$J_d(g_{\lambda_{\ell+1}}) < \alpha < J_d(g_{\lambda_\ell}). \tag{47}$$

For each p in the interval $[0, 1]$, define the Markov stationary policy f_p obtained by randomizing the policies g_{λ_ℓ} and $g_{\lambda_{\ell+1}}$ with bias p. Thus, the randomized policy f_p prescribes

Evict $doc(i)$ if

$$i = \begin{cases} \arg\min_{j \in S+r}(p(i)b_{\lambda_\ell}(i)) & \text{w.}p. \quad p \\ \arg\min_{j \in S+r}(p(i)b_{\lambda_{\ell+1}}(i)) & \text{w.}p. \quad 1-p. \end{cases} \tag{48}$$

Theorem 5: The optimal cache replacement policy $g^\star$ for the constrained caching problem is any randomized policy f_p of the form (48) with p determined through the saturation equation

$$J_d(f_p) = \alpha. \tag{49}$$

Proof. For the most part we follow the arguments of [3]: Let $\ell = 0, 1, \ldots, L-1$ be the interger appearing in the identification (46). Pick λ and μ in the *open* intervals $(\lambda_\ell, \lambda_{\ell+1})$ and $(\lambda_{\ell+1}, \lambda_{\ell+2})$, respectively, in which case

$$g_\lambda = g_{\lambda_\ell} \quad \text{and} \quad g_\mu = g_{\lambda_{\ell+1}}$$

with

$$J_d(g_\mu) < \alpha < J_d(g_\lambda). \tag{50}$$

Thus, as in the proof of Theorem 4.4 in [3], let λ and μ go to $\lambda_{\ell+1}$ monotonically under their respective constraints. The resulting limiting policies $\underline{g}$ and $\bar{g}$ (in the notation of [3]), are simply given here by

$$\underline{g} = g_{\lambda_{\ell+1}} \quad \text{and} \quad \bar{g} = g_{\lambda_\ell}$$

with [8]

$$J_\gamma(f_p) = J_\gamma(g_{\lambda_{\ell+1}}) = J_\gamma(g_{\lambda_\ell}) \tag{51}$$

for *every* p in the interval $[0, 1]$, and optimality

$$J_\gamma(f_p) \leq J_\gamma(\pi), \quad \pi \in \mathcal{P}$$

follows. Moreover, the mapping $p \to J_d(f_p)$ being continuous [12], with $J_d(f_p)_{p=0} = J_d(g_{\lambda_{\ell+1}})$ and $J_d(f_p)_{p=1} = J_d(g_{\lambda_\ell})$, there exists at least one value p in $(0, 1)$ such that (49) holds. The proof of optimality is now complete in view of

---

[8]See details in the proof of Theorem 4.4 in [3].

comments made at the beginning of Section V-C. ∎

It is possible to give a somewhat explicit expression for $J_d(f_p)$ using p in $[0, 1]$: Indeed, set

$$\begin{aligned} S^\star &:= S(\lambda_\ell) \cap S(\lambda_{\ell+1}) \\ &= \{\sigma_{\lambda_\ell}(1), \ldots, \sigma_{\lambda_\ell}(M-1)\}. \end{aligned}$$

Then, we have

$$\begin{aligned} J_d(f_p) &= \mathbf{E}[d(R)] \\ &\quad - \lim_{T \to \infty} \frac{1}{T+1} \mathbf{E}_{f_p}\left[\sum_{t=0}^{T} \mathbf{1}[R_t \in S_t] d(R_t)\right] \end{aligned}$$

with

$$\begin{aligned} &\lim_{T \to \infty} \frac{1}{T+1} \mathbf{E}_{f_p}\left[\sum_{t=0}^{T} \mathbf{1}[R_t \in S_t] d(R_t)\right] \\ &= \sum_{i \in S^\star} p(i)d(i) + r(p)p(\sigma_{\lambda_\ell}(M))d(\sigma_{\lambda_\ell}(M)) \\ &\quad + (1 - r(p))p(\sigma_{\lambda_\ell}(M+1))d(\sigma_{\lambda_\ell}(M+1)) \end{aligned}$$

where $r(p)$ represents the asymptotic fraction of time that the cache contains the document $\sigma_{\lambda_\ell}(M)$. It is a simple matter to check that

$$r(p) := \frac{p \cdot p(\sigma_{\lambda_\ell}(M))}{p \cdot p(\sigma_{\lambda_\ell}(M)) + (1-p) \cdot p(\sigma_{\lambda_\ell}(M+1))}.$$

Case 3 – Finally, assume that $J_d(g_\lambda) > \alpha$ for all $\lambda \geq 0$. This situation is of limited interest as we now argue: Fix $\lambda > 0$. For each policy π in $\mathcal{P}$, we can use the optimality of g_λ to write

$$\alpha < \lambda^{-1}J_c(g_\lambda) + J_d(g_\lambda) \leq \lambda^{-1}J_c(\pi) + J_d(\pi).$$

Thus, letting λ go to infinity, we conclude to

$$\alpha \leq J_d(\pi), \quad \pi \in \mathcal{P}.$$

The constrained caching problem has no feasible solution unless there exists a policy that saturates the constraint. Typically, the inequality above will be strict.

REFERENCES

[1] A. Aho, P. Denning and D. Ullman, "Principles of optimal page replacement," Journal of the ACM **18** (1971), pp. 80-93.
[2] O.I. Aven and E.G. Coffman and Y.A. Kogan, *Stochastic Analysis of Computer Storage*. D. Reidel Publishing Company, Dordrecht (Holland), 1987.
[3] F. Beutler and K. Ross, "Optimal Policies for controlled Markov chains with a constraint," J. Math. Analysis and Applications **112** (1985), pp. 236-252.
[4] P. Cao and S. Irani, "Cost-aware WWW proxy caching algorithms," In Proceedings of the 1997 USENIX Symposium on Internet Technology and Systems, Monterey (CA), December 197, pp. 193-206.
[5] E. Coffman and P. Denning, *Operating Systems Theory*, Prentice-Hall, NJ (1973).
[6] D. Heyman and M. Sobel, *Stochastic Models in Operations Research, Volume II: Stochastic Optimization*, McGraw-Hill, New York (NY), (1984).
[7] S. Hosseini-Khayat, "On optimal replacement of nonuniform cache objects," IEEE Transactions on Computers **COMP-49** (2000), pp. 769-778.

[8] S. Jin and A. Bestavros, "GreedyDual* Web caching algorithm: Exploiting the two sources of temporal locality in Web request streams," In Proceedings of the 5th International Web Caching and Content Delivery Workshop, Lisbon (Portugal), May 2000.

[9] S. Jin and A. Bestavros, "Popularity-aware GreedyDual-Size Web proxy caching algorithms," In Proceedings of ICDCS'2000: The IEEE International Conference on Distributed Computing Systems, Taipei (Taiwan), May 2000.

[10] S. Jin and A. Bestavros, "Sources and characteristics of Web temporal locality," In Proceedings of MASCOTS'2000: The IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, San Fransisco (CA), August 2000.

[11] S. Jin and A. Bestavros, "Temporal locality in Web request streams: Sources, characteristics, and caching implications" (Extended Abstract) In Proceedings of SIGMETRICS'2000: The ACM International Conference on Measurement and Modeling of Computer Systems, Santa Clara (CA), June 2000.

[12] D.-J. Ma, A.M. Makowski and A. Shwartz, "Stochastic approximations for finite–state Markov chains," Stochastic Processes and Their Applications **35** (1990), pp. 27-45.

[13] S.M. Ross, *Introduction to Stochastic Dynamic Programming*, Academic Press, New York (NY), (1984).

[14] D. Starobinski and D. Tse, "Probabilistic methods for Web caching," *Performance Evaluation* **46** (2001), pp. 125-137.

[15] N.E. Young, "On-line caching as cache size varies," In Proceedings of SODA'1991: The ACM-SIAM Symposium on Discrete Algorithms, San Francisco (CA), January 1991.

A. A PROOF OF THEOREM 1

Theorem 1 is a direct consequence of the following fact:
Proposition 1: For each $T = 0, 1, \ldots$, it holds that

$$\arg\min\{u \in S + r : \mathbf{E}[V_T(S + r - u, R^*)]\}$$
$$= \arg\min\{u \in S + r : p(u)c(u)\} \qquad (52)$$

for any state (S, r) in $\mathcal{X}^\star$ with r not in S.

Equality (52) is understood to mean that

$$\mathbf{E}[V_T(S + r - v, R^*)] = \min_{u \in S+r} \mathbf{E}[V_T(S + r - u, R^*)].$$

holds with v given by

$$v = \arg\min\{j \in S + r : p(j)c(j)\}. \qquad (53)$$

This statement is *weaker* than the monotonicity statement $p(u)c(u) \geq p(w)c(w)$ if

$$\mathbf{E}[V_T(S + r - u, R^*)]$$
$$\geq \mathbf{E}[V_T(S + r - w, R^*)], \quad u, w \in S + r.$$

which is used in the optimality proof of A_0 [1], [5].

The proof proceeds by induction on $T = 0, 1, \ldots$.
The basis step – Fix (S, r) in $\mathcal{X}^\star$ and note that

$$V_0(S, r) = \mathbf{1}[r \notin S]c(r).$$

Thus, for u in $S + r$ distinct from v (also in $S + r$ by virtue of its definition (53)), we have

$$\mathbf{E}[V_0(S + r - u, R^*)]$$
$$= \mathbf{E}[\mathbf{1}[R^* \notin S + r - u]c(R^*)]$$
$$= \mathbf{E}[\mathbf{1}[R^* \notin S + r]c(R^*)] + \mathbf{E}[\mathbf{1}[R^* = u]c(R^*)]$$

with a similar expression for $\mathbf{E}[V_0(S + r - v, R^*)]$. Hence,

$$\mathbf{E}[V_0(S + r - u, R^*)] - \mathbf{E}[V_0(S + r - v, R^*)]$$
$$= p(u)c(u) - p(v)c(v)$$

and (52) does hold for $T = 0$.

The induction step – Assume (52) to hold for some $T = 0, 1, \ldots$. Fix (S, r) in $\mathcal{X}^\star$ with r not in S. We need to show that for u in $S + r$, we have

$$\mathbf{E}[V_{T+1}(S + r - u, R^*) - V_{T+1}(S + r - v, R^*)] \geq 0 \quad (54)$$

with v given by (53).

Fix u in $S + r$ and let $R^{\star\star}$ denote an rv distributed like $R^\star$ and independent of it. Using the DPE (15) we can write

$$\mathbf{E}[V_{T+1}(S + r - u, R^*)] \qquad (55)$$
$$= \mathbf{P}[R^\star \in S + r - u]\mathbf{E}[V_T(S + r - u, R^{\star\star})]$$
$$+ \mathbf{E}[\mathbf{1}[R^\star \notin S + r - u]c(R^\star)]$$
$$+ \mathbf{E}\left[\mathbf{1}[R^\star \notin S + r - u]\tilde{V}_T(S + r - u, R^\star)\right]$$

with

$$\tilde{V}_T(S, x) := \min_{u' \in S+x} \mathbf{E}[V_T(S + x - u', R^{\star\star})]$$

for every set S with $|S| = M$ and x not in S.

Note that

$$\mathbf{P}[R^\star \in S + r - u]\mathbf{E}[V_T(S + r - u, R^{\star\star})]$$
$$= \mathbf{P}[R^\star \in S + r - (u, v)]\mathbf{E}[V_T(S + r - u, R^{\star\star})]$$
$$+ p(v)\mathbf{E}[V_T(S + r - u, R^{\star\star})] \qquad (56)$$

and that

$$\mathbf{E}[\mathbf{1}[R^\star \notin S + r - u]c(R^\star)]$$
$$= \mathbf{E}[\mathbf{1}[R^\star \notin S + r]c(R^\star)] + p(u)c(u) \qquad (57)$$

with v defined by (53). Finally,

$$\mathbf{E}\left[\mathbf{1}[R^\star \notin S + r - u]\tilde{V}_T(S + r - u, R^\star)\right]$$
$$= \mathbf{E}\left[\mathbf{1}[R^\star \notin S + r]\tilde{V}_T(S + r - u, R^\star)\right]$$
$$+ p(u)\tilde{V}_T(S + r - u, u). \qquad (58)$$

Reporting (56), (57) and (58) into (55), we conclude that

$$\mathbf{E}[V_{T+1}(S + r - u, R^*)]$$
$$= \mathbf{P}[R^\star \in S + r - (u, v)]\mathbf{E}[V_T(S + r - u, R^{\star\star})]$$
$$+ \mathbf{E}[\mathbf{1}[R^\star \notin S + r]c(R^\star)] + p(u)c(u)$$
$$+ p(u)\tilde{V}_T(S + r - u, u)$$
$$+ p(v)\mathbf{E}[V_T(S + r - u, R^{\star\star})]$$
$$+ \mathbf{E}\left[\mathbf{1}[R^\star \notin S + r]\tilde{V}_T(S + r - u, R^\star)\right]. \qquad (59)$$

We can now write the corresponding expression (59) with u replaced by v, and the difference in (54) takes the form

$$\mathbf{E}[V_{T+1}(S + r - u, R^*) - V_{T+1}(S + r - v, R^*)]$$
$$= (p(u)c(u) - p(v)c(v))$$
$$+ \mathbf{P}[R^\star \in S + r - (u, v)]\Delta_1$$
$$+ p(u)\Delta_2 + p(v)\Delta_3 + \Delta_4 \qquad (60)$$

with

$$\Delta_1 \quad := \quad \mathbf{E}\left[V_T(S+r-u, R^{\star\star})\right]$$
$$- \mathbf{E}\left[V_T(S+r-v, R^{\star\star})\right]$$

$$\Delta_2 := \tilde{V}_T(S+r-u, u) - \mathbf{E}\left[V_T(S+r-v, R^{\star\star})\right]$$

$$\Delta_3 := \mathbf{E}\left[V_T(S+r-u, R^{\star\star})\right] - \tilde{V}_T(S+r-v, v)$$

and

$$\Delta_4 \quad := \quad \mathbf{E}\left[\mathbf{1}\left[R^\star \notin S+r\right]\tilde{V}_T(S+r-u, R^\star)\right]$$
$$- \mathbf{E}\left[\mathbf{1}\left[R^\star \notin S+r\right]\tilde{V}_T(S+r-v, R^\star)\right].$$

Observe that $p(u)c(u) - p(v)c(v) \geq 0$ by the definition of v and that the condition $\Delta_1 \geq 0$, being equivalent to (52), holds true under the induction hypothesis. Next, we note that

$$\tilde{V}_T(S+r-u, u)$$
$$= \min_{u' \in S+r} \mathbf{E}\left[V_T(S+r-u', R^{\star\star})\right]$$
$$= \mathbf{E}\left[V_T(S+r-v, R^{\star\star})\right]$$

by the induction hypothesis and the definition of v, so that $\Delta_2 = 0$. Similarly,

$$\tilde{V}_T(S+r-v, v) = \min_{v' \in S+r} \mathbf{E}\left[V_T(S+r-v', R^{\star\star})\right]$$

whence $\Delta_3 \geq 0$ again by the induction hypothesis and the definition of v. Consequently, it is already the case that

$$\mathbf{E}\left[V_{T+1}(S+r-u, R^\star) - V_{T+1}(S+r-v, R^\star)\right] \geq \Delta_4$$

and (54)-(53) will hold if we can show that $\Delta_4 \geq 0$.

Inspection of Δ_4 reveals that $\Delta_4 \geq 0$ provided

$$\tilde{V}_T(S+r-u, x) - \tilde{V}_T(S+r-v, x) \geq 0 \qquad (61)$$

whenever x is *not* in $S+r$.

To establish (61) we find it useful to order the set of documents $\{1, \dots, M\}$ according to their expected cost: For u and v in $\{1, \dots, M\}$ we write $u < v$ (resp. $u \leq v$) if $p(u)c(u) < p(v)c(v)$ (resp. $p(u)c(u) \leq p(v)c(v)$), with equality $u = v$ if $p(u)c(u) = p(v)c(v)$. With this terminology we can now interpret v as the smallest element in $S+r$ according to this order. Two cases emerge depending on whether $v < x$ or $x \leq v$:

Case 1 – Assume $x \leq v$ with x is *not* in $S+r$. Then, for u in $S+r$ with $u \neq v$, we have

$$\tilde{V}_T(S+r-u, x) \qquad (62)$$
$$= \min_{u' \in S+r-u+x} \mathbf{E}\left[V_T(S+r-u+x-u', R^{\star\star})\right].$$

Note that x is not in $S+r-u$ and that x is smallest in $S+r+x$ (thus in $S+x-u$ which contains it). By the induction hypothesis applied in state $(S+r-u, x)$, the minimization in (62) is achieved at $u' = x$, so that

$$\tilde{V}_T(S+r-u, x) = \mathbf{E}\left[V_T(S+r-u, R^{\star\star})\right]. \qquad (63)$$

The same argument shows that

$$\tilde{V}_T(S+r-v, x)$$
$$= \min_{v' \in S+r-v+x} \mathbf{E}\left[V_T(S+r-v+x-v', R^{\star\star})\right]$$
$$= \mathbf{E}\left[V_T(S+r-v, R^{\star\star})\right] \qquad (64)$$

by applying the induction hypothesis in state $(S+r-v, x)$. Combining these facts, we get

$$\tilde{V}_T(S+r-u, x) - \tilde{V}_T(S+r-v, x) \qquad (65)$$
$$= \mathbf{E}\left[V_T(S+r-u, R^{\star\star})\right] - \mathbf{E}\left[V_T(S+r-v, R^{\star\star})\right]$$

and (61) follows by invoking the induction hypothesis once more, this time in state (S, r).

Case 2 – Assume $v < x$ with x is *not* in $S+r$. Then, going back to the expression (62) for $u \neq v$ in $S+r$, we note that now v is the smallest element of $S+r-u+x$, hence achieves the minimum in (62) by virtue of the induction hypothesis applied in state $(S+r-u, x)$. Therefore,

$$\tilde{V}_T(S+r-u, x)$$
$$= \mathbf{E}\left[V_T(S+r-u+x-v, R^{\star\star})\right]$$
$$= \mathbf{E}\left[V_T(S+r-v+x-u, R^{\star\star})\right]. \qquad (66)$$

On the other hand, by the induction hypothesis applied to the state $(S+r-v, x)$. we find that

$$\tilde{V}_T(S+r-v, x)$$
$$= \min_{v' \in S+r-v+x} \mathbf{E}\left[V_T(S+r-v+x-v', R^{\star\star})\right]$$
$$= \mathbf{E}\left[V_T(S+r-v+x-v^\star, R^{\star\star})\right] \qquad (67)$$

where $v^\star$ is the smallest element in $S+r-v+x$. Collecting these expressisons, we find

$$\tilde{V}_T(S+r-u, x) - \tilde{V}_T(S+r-v, x)$$
$$= \mathbf{E}\left[V_T(S+r-v+x-u, R^{\star\star})\right]$$
$$- \mathbf{E}\left[V_T(S+r-v+x-v^\star, R^{\star\star})\right] \qquad (68)$$

and (61) now follows by invoking the induction hypothesis once more, this time in state $(S+r-v, x)$, as we note that any element u in $S+r$ with $u \neq v$ is necessarily in $S+r-v$, hence in $S+r-v+x$. This completes the proof of Proposition 1. ∎