

# TECHNICAL RESEARCH REPORT

## Optimal Replacement Policies for Non-Uniform Cache Objects with Optional Eviction

*by Omri Bahat, Armand M. Makowski*

**CSHCN TR 2002-16  
(ISR TR 2002-28)**



*The Center for Satellite and Hybrid Communication Networks is a NASA-sponsored Commercial Space Center also supported by the Department of Defense (DOD), industry, the State of Maryland, the University of Maryland and the Institute for Systems Research. This document is a technical report in the CSHCN series originating at the University of Maryland.*

**Web site <http://www.isr.umd.edu/CSHCN/>**

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

Omri Bahat, Armand M. Makowski

**Abstract**—Replacement policies for general caching applications and Web caching in particular have been extensively addressed in the literature. Many policies that focus on document costs, size, probability of references and temporal locality of requested documents, have been proposed. In many cases these policies are ad-hoc attempts to take advantage of the statistical information contained in the stream of requests, and to address the factors above. However, since the introduction of optimal replacement policies for conventional caching, the problem of finding optimal replacement policies under the factors indicated has not been studied 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  $C_0$ , minimizes the long-run average metric induced by non-uniform document costs when document eviction is optional. We then use these results to propose a framework to operate 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  $C_0$  but induced by different costs.

**Index Terms**—Web caching, Optimal replacement policies, Non-uniform cost, Independent Reference Model, Caching under a constraint, Markov decision processes.

## 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 ra-

The authors are with the Department of Electrical and Computer Engineering, and the Institute for Systems Research, University of Maryland, College Park, MD 20742. E-mail: obahat@glue.umd.edu, armand@isr.umd.edu.

This material is based upon work supported by the Space and Naval Warfare Systems Center – San Diego under Contract No. N66001-00-C-8063. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Space and Naval Warfare Systems Center – San Diego.

tios. A large number of techniques for file caching and virtual memory replacement have been developed [1] [4], but unfortunately they do not necessarily transfer to Web caching as explained below. Despite the ever decreasing prices 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] [4] 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 occurrence (equivalently, is the least popular) among the documents in the cache. More precisely, let  $doc(1), \dots, doc(N)$  denote the set of documents to be requested and let  $p(j)$  denote the probability of reference of  $doc(j)$  ( $j = 1, \dots, N$ ). When the cache is full and the requested document is not in the cache,  $A_0$  prescribes

$$\begin{aligned} & \text{Evict } doc(i) \\ & \text{if } i = \arg \min (p(j) : doc(j) \text{ in cache}). \end{aligned} \quad (1)$$

In practice, the popularity vector  $\mathbf{p} = (p(1), \dots, p(N))$  is not available and thus needs to be estimated on-line as requests are coming in. 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  $K^{th}$  requested document is not in the cache, LFU prescribes

$$\begin{aligned} & \text{Evict } doc(i) \\ & \text{if } i = \arg \min (\hat{p}_K(j) : doc(j) \text{ in cache}) \end{aligned} \quad (2)$$

where  $\hat{p}_K(j)$  is the frequency estimate of  $p(j)$  based on the trace measurements up to the  $K^{th}$  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 the object. 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 each 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, pre-

clude an easy transfer of caching techniques developed earlier for computer system memory. Yet, a large number of studies have focused on the design of efficient replacement policies; see [6] [7] [8] [9] 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 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] and [4] is too restrictive to capture the salient features present in Web caching. Indeed, the Independent Reference Model fails to 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,<sup>1</sup> 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 these provably optimal caching policies in several directions: (i) The documents have non-uniform costs (as we assimilate cost to size and variable retrieval latency), (ii) there exist correlations in the request streams, and (iii) document placement and replacement are 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  $c(j)$  ( $j = 1, \dots, N$ ) 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 optimal replacement policy for the minimum average cost criterion (over both finite and infinite horizons). Making use of standard ideas from the theory of Markov Decision Processes (MDPs), we show [Theorem 1] that the optimal policy is

<sup>1</sup>In web caching timescales are slower than in conventional caching due to variable network latencies.

the (pure) Markov stationary policy  $C_0$  that prescribes

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

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

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

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

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

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  $C_0$  as given in (3).

The ability to find provably optimal policies under an arbitrary cost structure can be put to advantage in the following way: 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 on retrieved documents while the former does not. One possible approach to capture the multi-criteria aspects is to introduce constraints. In the second part of the paper we formulate the problem of finding a constrained replacement policy that minimizes an average cost under a single constraint in terms of a long-run average metric. Using the developments indicated above we are able to 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 [2].

The paper is organized as follows: The search for optimal replacement policies with optional eviction is formulated as a Markov decision process in Section II. Its solution is discussed in Section III and Section IV is devoted to the constrained problem.

## II. FINDING GOOD REPLACEMENT POLICIES

One approach for designing good replacement policies is to couch the problem as one of sequential decision making in the presence of randomness. The analysis that produced the policy  $A_0$  described earlier (and its optimality under the Independent Reference Model) is one based on Dynamic Programming as developed in the framework of MDPs [5] [11].

### A. An MDP framework

The system is composed of a server where a copy of each of its  $N$  documents is available, and of 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. In principle this decision is taken on the basis of earlier decisions and past requests, and seeks to minimize a cost function associated with the operation of the cache over either a finite horizon or an infinite horizon.

Decision epochs are defined as the instants at which requests for documents are presented at the cache, and are indexed by  $t = 0, 1, \dots$ . At time  $t = 0, 1, \dots$ , let  $S_t$  denote the state of the cache, thus  $S_t$  is a subset of  $\{1, \dots, N\}$  with size  $|S_t| \leq M$ . Let  $\mathcal{S}_M$  denote the collection of all subsets of  $\{1, \dots, N\}$  of size less or equal to  $M$ .

Let  $\{R_t, t = 0, 1, \dots\}$  denote the sequence of document requests, with  $R_t$  an  $\{1, \dots, N\}$ -valued random variable (rv). 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<sup>2</sup>

$$S_{t+1} = T(S_t, R_t, U_t)$$

<sup>2</sup>Throughout, for any subset  $S$  of  $\{1, \dots, N\}$  and any elements  $x$  and  $u$  in  $\{1, \dots, N\}$ , we write  $S + x - u$  to denote the subset of  $\{1, \dots, N\}$  obtained from  $S$  by adding  $x$  to it and removing  $u$  from the resulting set, in that order.

$$= \begin{cases} S_t & \text{if } R_t \in S_t \\ S_t + R_t & \text{if } R_t \notin S_t, |S_t| < M \\ S_t + R_t - U_t & \text{if } R_t \notin S_t, |S_t| = M \end{cases}$$

In this formulation we note that a document is not necessarily evicted from the cache if the requested document is not in cache and the cache is full. Under these rules of operation, we note that eventually the cache will become full at some time and will remain so from that time onward, i.e., given any initial cache  $S_0$ , there exists  $\tau$  finite such that  $|S_{\tau+t}| = M$  for all  $t = 0, 1, \dots$

The state variable at time  $t = 0, 1, \dots$  is the pair  $(S_t, R_t)$ . The state space is then the set  $\mathcal{X}$  given by

$$\mathcal{X} := \mathcal{S}_M \times \{1, \dots, N\}.$$

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

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

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

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

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

$$U_t = \pi_t(H_t)$$

for some mapping  $\pi_t : \mathcal{H}_t \rightarrow \{0, 1, \dots, N\}$ . The collection  $\pi = (\pi_t, t = 0, 1, \dots)$  defines the replacement (or eviction) policy  $\pi$ . Sometimes it is useful to consider randomized policies which are now defined: A randomized replacement policy  $\pi$  is a collection  $(\pi_t, t = 0, 1, \dots)$  of mappings  $\pi_t : \{0, 1, \dots, N\} \times \mathcal{H}_t \rightarrow [0, 1]$  such that for all  $t = 0, 1, \dots$ , we have

$$\sum_{u=0}^N \pi(u; H_t) = 1$$

with

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

and

$$\pi(u; H_t) = \delta(u, 0), \quad R_t \in S_t \text{ or } |S_t| < M$$

for all  $u = 0, 1, \dots, N$ . The class of all (possibly randomized) replacement policies is denoted by  $\mathcal{P}$ .

If the replacement policy  $\pi$  has the property that

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

for mappings  $f_t : \mathcal{X} \rightarrow \{0, 1, \dots, N\}$ , we say that  $\pi$  is a Markov policy. If in addition,  $f_t = f$  for all  $t = 0, 1, \dots$  the policy is said to be a (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 [5].

Under the Independent Reference Model, the sequence of requests is a sequence  $\{R_t, t = 0, 1, \dots\}$  of  $\{1, \dots, N\}$ -valued rv distributed according to some pmf  $\mathbf{p} = (p(1), \dots, p(N))$  on  $\{1, \dots, N\}$ .

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

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

and

$$\begin{aligned} \mathbf{P}_\pi [S_{t+1} = S', R_{t+1} = y | H_t, U_t] & \quad (6) \\ &= p(y) \mathbf{P}_\pi [S_{t+1} = S' | H_t, U_t] \\ &= p(y) \mathbf{1} [T(S_t, R_t, U_t) = S'] \end{aligned}$$

for every state  $(S', y)$  in  $\mathcal{X}$ . 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  $c : \{1, \dots, N\} \rightarrow \mathbb{R}_+$ , we associate several cost functions: Fix a replacement policy  $\pi$  in  $\mathcal{P}$ . For each  $T = 0, 1, \dots$ , define the total cost over the horizon  $[0, T]$  under the policy  $\pi$  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  $\pi$  is then defined by

$$\begin{aligned} J_c(\pi) &= \limsup_{T \rightarrow \infty} \frac{1}{T+1} J_c(\pi; T) \quad (7) \\ &= \limsup_{T \rightarrow \infty} \frac{1}{T+1} \mathbf{E}_\pi \left[ \sum_{t=0}^T \mathbf{1} [R_t \notin S_t] c(R_t) \right] \end{aligned}$$

We use the limsup operation in the definition above since under an arbitrary policy  $\pi$  the limit in (7) may not exist; this is standard practice in the theory of MDPs.

A number of situations can be handled by adequately specializing the cost-per-step  $c$ : Indeed, if  $c(y) = 1$  ( $y = 1, \dots, N$ ), 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  $\pi$ , respectively.

On the other hand, if  $c$  is taken to be the size function  $s : \{1, \dots, N\} \rightarrow \mathbb{N}$ , with  $s(y)$  denoting the size (in bytes) of  $doc(y)$  ( $y = 1, \dots, N$ ), then the byte hit rate under policy  $\pi$  can be defined by

$$BHR(\pi) = \liminf_{T \rightarrow \infty} \frac{\mathbf{E}_\pi \left[ \sum_{t=0}^T \mathbf{1}[R_t \in S_t] s(R_t) \right]}{\mathbf{E}_\pi \left[ \sum_{t=0}^T s(R_t) \right]} \quad (8)$$

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

$$\lim_{T \rightarrow \infty} \frac{1}{T+1} \mathbf{E} \left[ \sum_{t=0}^T s(R_t) \right] = \mathbf{E} [s(R)]$$

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

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

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

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

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

We refer to any such policy  $\pi^*$  as an optimal replacement policy. It is not necessarily unique, but in the next section we identify such an optimal policy  $\pi^*$  which also happens to be a Markov stationary policy.

### 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* mandated. A useful characterization of the optimal policy for the corresponding MDP (being one with finite state and action spaces) can be initiated with the help of the Dynamic Programming Equation (DPE) [11].

#### A. The optimal replacement policy

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

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

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

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

Recall that regardless of the initial condition, the cache will eventually become and remain full. Thus, under the average cost criterion used here, there is no loss of generality in assuming the space state to be  $\mathcal{X}^*$  (instead of the original  $\mathcal{X}$ ) with

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

For the MDP at hand, the DPE takes the form

$$\begin{aligned} &V_{T+1}(S, r) \\ &= \mathbf{1}[r \in S] \mathbf{E} [V_T(S, R^*)] \\ &+ \mathbf{1}[r \notin S] \left( c(r) + \min_{u \in S+r} \mathbf{E} [V_T(S+r-u, R^*)] \right) \end{aligned} \quad (9)$$

for every state  $(S, r)$  in  $\mathcal{X}^*$  with  $R^*$  denoting an  $\{1, \dots, N\}$ -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 [11], 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

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

say with a lexicographic tie-breaker for sake of concreteness.

The main result of this section is contained in the following theorem that prescribes the optimal replacement policy for the caching problem at hand.

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

$$g_T^*(S, r) = g_*(S, r) \quad (10)$$

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

$$g_*(S, r) := \arg \min_{u \in S+r} (p(u)c(u)). \quad (11)$$

The proof of Theorem 1 is given in Appendix A. Note that  $g_T^*$  does not depend on  $T$ , and that the Markov stationary policy associated with  $g_*$  is the policy  $C_0$  introduced earlier. It is now plain from Theorem 1 that the Markov stationary policy  $C_0$  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  $\sigma$  of  $\{1, \dots, N\}$  which orders the values  $p(i)c(i)$  ( $i = 1, \dots, N$ ) in decreasing order, namely

$$p(\sigma(1))c(\sigma(1)) \geq p(\sigma(2))c(\sigma(2)) \geq \dots \quad (12)$$

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

$$S := \{\sigma(1), \dots, \sigma(M)\} \quad (13)$$

for this steady-state stack, then formally

$$\lim_{t \rightarrow \infty} \mathbf{P}_{C_0} [\sigma(i) \in S_t] = 0, \quad i = M + 1, \dots, N$$

and

$$J_d(C_0) = \sum_{i \notin S} p(i)d(i) = \sum_{i=M+1}^N p(\sigma(i))d(\sigma(i)) \quad (14)$$

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

$$BHR(C_0) = \frac{\sum_{j=1}^M p(\sigma(j))s(\sigma(j))}{\mathbf{E}[s(R)]}. \quad (15)$$

Another interesting observation is the relation of the optimal replacement policy  $C_0$  to the well-established Greedy Dual\* and Greedy Dual-Size replacement policies described in [6] and [7]. Let  $c_{GD} : \{1, \dots, N\} \rightarrow \mathbb{R}_+$  be an arbitrary cost used by the Greedy Dual policies. The Greedy Dual policies under optional document placement in case of a cache miss prescribe

$$\begin{aligned} & \text{Evict } doc(i) \\ & \text{if } i = \arg \min_{j \in S+r} \left( L + \left( \frac{p(j)c_{GD}(j)}{s(j)} \right)^{\frac{1}{\beta}} \right) \end{aligned} \quad (16)$$

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 used here, the temporal locality factor  $L$  can be taken to be zero, in which case the Greedy Dual policy simplifies to

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

which is a special case of the optimal replacement policy  $C_0$  associated with cost function  $c : \{1, \dots, N\} \rightarrow \mathbb{R}_+$  given by

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

### C. Implementing the optimal policy

A natural implementation of the optimal replacement policy  $C_0$  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 ( $c(j)$ ,  $j = 1, \dots, N$ ), 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  $(\hat{c}_K(j), j = 1, \dots, N)$  denote an estimate of the document costs which are available at the cache at the time instance of the  $K^{\text{th}}$  request: If  $|S_t| < M$ , document placement always takes place; otherwise the replacement action is dictated by

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

## IV. 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  $c, d : \{1, \dots, N\} \rightarrow \mathbb{R}_+$ . 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

$$c(y) = 1 \quad \text{and} \quad d(y) = s(y), \quad y = 1, \dots, N \quad (17)$$

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  $\pi$  in  $\mathcal{P}$  satisfies the constraint at level  $\alpha$  if

$$J_d(\pi) \leq \alpha. \quad (18)$$

Let  $\mathcal{P}(d; \alpha)$  denote the class of all cache replacement policies in  $\mathcal{P}$  that satisfy the constraint (18) at level  $\alpha$ .

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

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

We refer to any such policy  $\pi^*$  as a constrained optimal policy (at level  $\alpha$ ). With the choice (17) 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

$$\tilde{J}_\lambda(\pi) = J_c(\pi) + \lambda J_d(\pi), \quad \pi \in \mathcal{P}, \lambda \geq 0. \quad (19)$$

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

$$\tilde{J}_\lambda(\pi^*(\lambda)) \leq \tilde{J}_\lambda(\pi), \quad \pi \in \mathcal{P}. \quad (20)$$

Now, if for some  $\lambda^* \geq 0$ , the policy  $\pi^*(\lambda^*)$  happens to saturate the constraint at level  $\alpha$ , i.e.,

$$J_d(\pi^*(\lambda^*)) = \alpha,$$

then, the optimality of  $\pi^*(\lambda^*)$  implies

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

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

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

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

The only glitch in this approach resides in the use of the limsup operation in the definition (7), so that  $\tilde{J}_\lambda(\pi)$  is not necessarily the long-run average cost under policy  $\pi$  for some appropriate one-step cost. Thus, finding the optimal cache replacement policy  $\pi^*(\lambda)$  specified by (20) cannot be achieved in a straightforward manner.

## B. A Lagrangian approach

Following the treatment in [2], 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  $b_\lambda : \{1, \dots, N\} \rightarrow \mathbb{R}_+$  by

$$b_\lambda(y) := c(y) + \lambda d(y), \quad y = 1, \dots, N$$

and consider the corresponding long-run average functional (7), i.e., for any policy  $\pi$  in  $\mathcal{P}$ , we set

$$\begin{aligned} J_\lambda(\pi) &:= J_{b_\lambda}(\pi) \\ &= \limsup_{T \rightarrow \infty} \frac{1}{T+1} \mathbf{E}_\pi \left[ \sum_{t=0}^T \mathbf{1}[R_t \notin S_t] b_\lambda(R_t) \right]. \end{aligned} \quad (21)$$

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  $\pi$  is a Markov stationary policy.

For each  $\lambda \geq 0$ , the (unconstrained) caching problem associated with the cost  $b_\lambda$  is an MDP with finite state and action spaces. Thus, there exists a Markov stationary policy, denoted  $g_\lambda$ , which is optimal, 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  $g_\lambda$  also minimizes the Lagrangian functional (19), and the relation

$$J_\lambda(g_\lambda) = \inf_{\pi \in \mathcal{P}} J_\lambda(\pi) = \inf_{\pi \in \mathcal{P}} \tilde{J}_\lambda(\pi) \quad (22)$$

holds. Consequently, as argued in Section IV-A, if for some  $\lambda^* \geq 0$ , the policy  $g_{\lambda^*}$  saturates the constraint at level  $\alpha$ , then the policy  $g_{\lambda^*}$  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 Markov stationary policy  $g^*$  such that

$$J_d(g^*) = \alpha \quad (23)$$

and

$$J_\gamma(g^*) \leq J_\gamma(\pi), \quad \pi \in \mathcal{P}. \quad (24)$$



### C. On the way to solving the constrained MDP

The appropriate multiplier  $\gamma$  and the policy  $g^*$  appearing in (23) and (24) will be identified in Section IV-D. To help us in this process we need some technical facts and notation which we now develop.

*Theorem 2: The optimal cost function  $\lambda \rightarrow J_\lambda(g_\lambda)$  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  $g_\lambda$  as the policy  $C_0$  induced by  $b_\lambda$ , i.e.,

$$\text{Evict } doc(i) \quad \text{iff} \quad i = \arg \min_{j \in S^{+r}} (c(j) + \lambda d(j)). \quad (25)$$

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

$$p(\sigma_\lambda(1))b_\lambda(\sigma_\lambda(1)) \geq p(\sigma_\lambda(2))b_\lambda(\sigma_\lambda(2)) \geq \dots \quad (26)$$

with a lexicographic tie-breaker. Let  $S(\lambda)$  denote the steady-state stack induced by the policy  $g_\lambda$ , namely the collection of documents in the cache that results from long-term usage of the policy  $g_\lambda$ . Obviously, we have<sup>3</sup>

$$S(\lambda) = \{\sigma_\lambda(1), \dots, \sigma_\lambda(M)\} \quad (27)$$

so that

$$J_\lambda(g_\lambda) = J_{b_\lambda}(g_\lambda) = \sum_{i \notin S(\lambda)} p(i)b_\lambda(i) \quad (28)$$

upon rephrasing comments made earlier in Section III.

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

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

In view of (28) it is plain that

$$J_\lambda(g_\lambda) = \sum_{i \notin S(\lambda_\ell)} p(i)b_\lambda(i) \quad (29)$$

whenever  $\lambda$  belongs to  $I_\ell$  for some  $\ell = 0, \dots, L$ .

<sup>3</sup>The steady-state stack  $S$  given by (13) corresponds to the case  $\lambda = 0$  with  $\sigma = \sigma_0$ .

**Proof.** For each policy  $\pi$  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  $\lambda \rightarrow \tilde{J}_\lambda(\pi)$  is non-decreasing and affine, and we conclude from (22) that the mapping  $\lambda \rightarrow J_\lambda(g_\lambda)$  is indeed non-decreasing and concave. Its piecewise-linear character is a straightforward consequence of (29). ■

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  $i, j = 1, \dots, N$ , then there does not exist any  $k \neq i, j$  with  $k = 1, \dots, N$  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  $\ell = 0, 1, \dots, L$ , the relative position of the quantities  $p(i)b_\lambda(i)$  ( $i = 1, \dots, N$ ) remains unchanged as  $\lambda$  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

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

and

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

By continuity we must have

$$\begin{aligned} & p(\sigma_{\lambda_\ell}(M))b_{\lambda_{\ell+1}}(\sigma_{\lambda_\ell}(M)) \\ &= p(\sigma_{\lambda_{\ell+1}}(M+1))b_{\lambda_{\ell+1}}(\sigma_{\lambda_\ell}(M+1)). \end{aligned} \quad (30)$$

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

**Proof.** The analog of (29) holds in the form

$$J_d(g_\lambda) = \sum_{i \notin S(\lambda_\ell)} p(i)d(i) \quad (31)$$

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

Now pick  $\ell = 0, 1, \dots, L-1$  and consider  $\lambda$  and  $\mu$  in the *open* intervals  $(\lambda_\ell, \lambda_{\ell+1})$  and  $(\lambda_{\ell+1}, \lambda_{\ell+2})$ , respectively. The desired monotonicity will be established if we

can show that  $J_d(g_\mu) - J_d(g_\lambda) \leq 0$ . First, from (31), we note

$$\begin{aligned} & J_d(g_\mu) - J_d(g_\lambda) \\ &= \sum_{i \in S(\lambda)} p(i)d(i) - \sum_{i \in S(\mu)} p(i)d(i) \\ &= p(\sigma_{\lambda_\ell}(M))d(\sigma_{\lambda_\ell}(M)) \\ &\quad - p(\sigma_{\lambda_\ell}(M+1))d(\sigma_{\lambda_\ell}(M+1)) \end{aligned} \quad (32)$$

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 (29) we get  $S(\lambda + \varepsilon) = S(\lambda)$  and

$$\begin{aligned} & J_{\lambda+\varepsilon}(g_{\lambda+\varepsilon}) - J_\lambda(g_\lambda) \\ &= \sum_{i \notin S(\lambda+\varepsilon)} p(i)b_{\lambda+\varepsilon}(i) - \sum_{i \notin S(\lambda)} p(i)b_\lambda(i) \\ &= \sum_{i \notin S(\lambda)} p(i)b_{\lambda+\varepsilon}(i) - \sum_{i \notin S(\lambda)} p(i)b_\lambda(i) \\ &= \varepsilon \sum_{i \notin S(\lambda)} p(i)d(i) \end{aligned} \quad (33)$$

Similarly,

$$J_{\mu+\varepsilon}(g_{\mu+\varepsilon}) - J_\mu(g_\mu) = \varepsilon \sum_{i \notin S(\mu)} p(i)d(i). \quad (34)$$

By Theorem 2, the mapping  $\lambda \rightarrow J_\lambda(g_\lambda)$  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 (33) and (34) in this inequality, we readily conclude that

$$\sum_{i \in S(\lambda)} p(i)d(i) \leq \sum_{i \in S(\mu)} p(i)d(i). \quad (35)$$

But  $S(\lambda) = S(\lambda_\ell)$  and  $S(\mu) = S(\lambda_{\ell+1})$ , whence (35) 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 (32).  $\blacksquare$

#### 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  $g^*$  is simply the optimal replacement policy  $C_0$  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  $g_\lambda$  above saturates the constraint at level  $\alpha$  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  $g_\lambda$  can be taken to be the optimal replacement policy  $g^*$  for the constrained caching problem (and the constraint is saturated).

**Case 2b** – The case of greater 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

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

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

$$J_d(g_{\lambda_{\ell+1}}) < \alpha < J_d(g_{\lambda_\ell}) \quad (36)$$

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

$$\begin{aligned} & \text{Evict } doc(i) \text{ if} \\ & i = \begin{cases} \arg \min_{j \in S+r} (p(i)b_{\lambda_\ell}(i)) & \text{w.p. } p \\ \arg \min_{j \in S+r} (p(i)b_{\lambda_{\ell+1}}(i)) & \text{w.p. } 1-p \end{cases} \end{aligned} \quad (37)$$

*Theorem 5:* The optimal cache replacement policy  $g^*$  for the constrained caching problem is any randomized policy  $f_p$  of the form (37) with  $p$  determined through the saturation equation

$$J_d(f_p) = \alpha. \quad (38)$$

**Proof.** For the most part we follow the arguments of

[2]: Pick  $\lambda$  and  $\mu$  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

$$J_d(g_\mu) < \alpha < J_d(g_\lambda). \quad (39)$$

Thus, as in the proof of Theorem 4.4 in [2], let  $\lambda$  and  $\mu$  go to  $\lambda_{\ell+1}$  monotonically under their respective constraints. The resulting limiting policies  $\underline{g}$  and  $\bar{g}$  (in the notation of [2]), are simply given here by

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

with <sup>4</sup>

$$J_\gamma(f_p) = J_\gamma(g_{\lambda_{\ell+1}}) = J_\gamma(g_{\lambda_\ell}) \quad (40)$$

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  $p \rightarrow J_d(f_p)$  being continuous [10], with

$$J_d(f_p)_{p=0} = J_d(g_{\lambda_{\ell+1}}) \quad \text{and} \quad J_d(f_p)_{p=1} = J_d(g_{\lambda_\ell}),$$

there exists at least one value  $p$  in  $(0, 1)$  such that (38) holds. The proof of optimality is now complete in view of comments made at the beginning of Section IV-C. ■

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

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

Then, we have

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

with

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

<sup>4</sup>See details in the proof of Theorem 4.4 in [2].

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

$$J_d(g_\lambda) > \alpha, \quad \lambda \geq 0.$$

This situation is of limited interest as we now argue: Fix  $\lambda > 0$ . For each policy  $\pi$  in  $\mathcal{P}$ , we can use the optimality of  $g_\lambda$  to write

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

Thus, letting  $\lambda$  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,” J. ACM, vol.18, No. 1, January 1971.
- [2] F. Beutler and K. Ross, “Optimal Policies for Controlled Markov Chains with a Constraint,” J. Math. Analysis and Applications 112(1985), pp. 236-252.
- [3] P. Cao and S. Irani, “Cost-Aware WWW Proxy Caching Algorithms,” In Proceedings of the 1997 USENIX Symposium on Internet Technology and Systems, pp. 193-206, Dec 1997.
- [4] E. Coffman and P. Denning, *Operating Systems Theory*, Prentice-Hall, NJ (1973).
- [5] D. Heyman and M. Sobel, *Stochastic Models in Operations Research, Volume II: Stochastic Optimization*, McGraw-Hill, New York (NY), (1984).
- [6] 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.
- [7] 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, Taiwan, May 2000.
- [8] 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 Francisco, CA, August 2000.
- [9] 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.

- [10] D.-J. Ma, A.M. Makowski and A. Shwartz, "Stochastic approximations for finite-state Markov chains," *Stochastic Processes and Their Applications*, vol. 35 (1990), pp. 27-45.
- [11] S.M. Ross, *Introduction to Stochastic Dynamic Programming*, Academic Press, New York (NY), (1984).
- [12] N.E. Young, "On-line caching as cache size varies," In *Proceedings of Symposium on Discrete Algorithms*, January 1991.

### A. A PROOF OF THEOREM 1

The optimality of the Markov stationary policy  $C_0$  for both finite and infinite horizon cost problems is a direct consequence of the following fact:

*Proposition 1:* For each  $T = 0, 1, \dots$ , it holds that

$$\begin{aligned} & \arg \min \{u \in S + r : \mathbf{E} [V_T(S + r - u, R^*)]\} \\ &= \arg \min \{u \in S + r : p(u)c(u)\} \end{aligned} \quad (41)$$

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

The equality (41) is understood to mean that if  $v := \arg \min \{u \in S + r : p(u)c(u)\}$ , then

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

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

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

which is in the optimality proof in [1], [4].

The proof of Proposition 1 proceeds by induction on  $T = 0, 1, \dots$

**The basis step** – Fix  $(S, r)$  in  $\mathcal{X}^*$  and note that

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

Thus, for distinct  $u$  and  $v$  in  $S + r$ , we have

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

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

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

and (41) does hold for  $T = 0$ .

**The induction step** – Assume (41) to hold for some  $T = 0, 1, \dots$ . Fix  $(S, r)$  in  $\mathcal{X}^*$  with  $r$  not in  $S$ . We need to show that for  $u$  in  $S + r$ , we have

$$\mathbf{E} [V_{T+1}(S + r - u, R^*)] - V_{T+1}(S + r - v, R^*) \geq 0 \quad (42)$$

if

$$v = \arg \min \{j \in S + r : p(j)c(j)\}. \quad (43)$$

Fix  $u$  in  $S + r$  and let  $R^{**}$  denote an rv distributed like  $R^*$  and independent of it. Using the DPE (9) we can write

$$\begin{aligned} & \mathbf{E} [V_{T+1}(S + r - u, R^*)] \\ &= \mathbf{P} [R^* \in S + r - u] \mathbf{E} [V_T(S + r - u, R^{**})] \\ & \quad + \mathbf{E} [\mathbf{1} [R^* \notin S + r - u] c(R^*)] \\ & \quad + \mathbf{E} [\mathbf{1} [R^* \notin S + r - u] \tilde{V}_T(S + r - u, R^*)] \end{aligned} \quad (44)$$

with

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

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

Note that

$$\begin{aligned} & \mathbf{P} [R^* \in S + r - u] \mathbf{E} [V_T(S + r - u, R^{**})] \\ &= \mathbf{P} [R^* \in S + r - (u, v)] \mathbf{E} [V_T(S + r - u, R^{**})] \\ & \quad + p(v) \mathbf{E} [V_T(S + r - u, R^{**})] \end{aligned} \quad (45)$$

and that

$$\begin{aligned} & \mathbf{E} [\mathbf{1} [R^* \notin S + r - u] c(R^*)] \\ &= \mathbf{E} [\mathbf{1} [R^* \notin S + r] c(R^*)] + p(u)c(u) \end{aligned} \quad (46)$$

with  $v$  as defined by (43). Finally,

$$\begin{aligned} & \mathbf{E} [\mathbf{1} [R^* \notin S + r - u] \tilde{V}_T(S + r - u, R^*)] \\ &= \mathbf{E} [\mathbf{1} [R^* \notin S + r] \tilde{V}_T(S + r - u, R^*)] \\ & \quad + p(u) \tilde{V}_T(S + r - u, u) \end{aligned} \quad (47)$$

Reporting (45), (46) and (47) into (44), we conclude that

$$\begin{aligned} & \mathbf{E} [V_{T+1}(S + r - u, R^*)] \\ &= \mathbf{P} [R^* \in S + r - (u, v)] \mathbf{E} [V_T(S + r - u, R^{**})] \\ & \quad + \mathbf{E} [\mathbf{1} [R^* \notin S + r] c(R^*)] + p(u)c(u) \\ & \quad + p(u) \tilde{V}_T(S + r - u, u) \\ & \quad + p(v) \mathbf{E} [V_T(S + r - u, R^{**})] \\ & \quad + \mathbf{E} [\mathbf{1} [R^* \notin S + r] \tilde{V}_T(S + r - u, R^*)] \end{aligned} \quad (48)$$

We can now write the corresponding expression (48) with  $u$  replaced by  $v$ , and the difference in (42) takes the form

$$\begin{aligned} & \mathbf{E} [V_{T+1}(S + r - u, R^*)] - V_{T+1}(S + r - v, R^*) \\ &= (p(u)c(u) - p(v)c(v)) \\ & \quad + \mathbf{P} [R^* \in S + r - (u, v)] \Delta_1 \\ & \quad + p(u)\Delta_2 + p(v)\Delta_3 + \Delta_4 \end{aligned} \quad (49)$$

with

$$\begin{aligned}\Delta_1 &:= \mathbf{E} [V_T(S+r-u, R^{**})] \\ &\quad - \mathbf{E} [V_T(S+r-v, R^{**})]\end{aligned}$$

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

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

and

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

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 (41), holds true under the induction hypothesis. Next, we note that

$$\begin{aligned}\tilde{V}_T(S+r-u, u) &= \min_{u' \in S+r} \mathbf{E} [V_T(S+r-u', R^{**})] \\ &= \mathbf{E} [V_T(S+r-v, R^{**})]\end{aligned}$$

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

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

whence  $\Delta_3 \geq 0$  again by the induction hypothesis and the definition of  $v$ . Consequently,

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

and (42)-(43) will hold if we can show that  $\Delta_4 \geq 0$ .

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

$$\tilde{V}_T(S+r-u, x) - \tilde{V}_T(S+r-v, x) \geq 0 \quad (50)$$

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

To establish (50) 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  $p(u)c(u) = p(v)c(v)$ . We can now interpret  $v$  as the smallest element in  $S+r$  according to this order. Two cases emerge depending on whether  $v \leq x$  or  $x < v$ :

**Case 1** – Assume  $x \leq v$ . Then, for  $u \neq v$  in  $S+r$ , we have

$$\begin{aligned}\tilde{V}_T(S+r-u, x) &= \min_{u' \in S+r-u+x} \mathbf{E} [V_T(S+r-u+x-u', R^{**})]\end{aligned} \quad (51)$$

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 to the state  $(S+r-u, x)$ ) we can conclude that the minimization above is achieved at  $u' = x$ , so that

$$\tilde{V}_T(S+r-u, x) = \mathbf{E} [V_T(S+r-u, R^{**})] \quad (52)$$

The same argument shows that

$$\begin{aligned}\tilde{V}_T(S+r-v, x) &= \min_{v' \in S+r-v+x} \mathbf{E} [V_T(S+r-v+x-v', R^{**})] \\ &= \mathbf{E} [V_T(S+r-v, R^{**})]\end{aligned} \quad (53)$$

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

$$\begin{aligned}\tilde{V}_T(S+r-u, x) - \tilde{V}_T(S+r-v, x) &= \mathbf{E} [V_T(S+r-u, R^{**})] - \mathbf{E} [V_T(S+r-v, R^{**})]\end{aligned} \quad (54)$$

and (50) follows by invoking the induction hypothesis once more, this time in state  $(S, r)$ .

**Case 2** – Assume  $v < x$ . Then, going back to the expression (51) 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 (51) by virtue of the induction hypothesis applied to the state  $(S+r-u, x)$ . Therefore,

$$\begin{aligned}\tilde{V}_T(S+r-u, x) &= \mathbf{E} [V_T(S+r-u+x-v, R^{**})] \\ &= \mathbf{E} [V_T(S+r-v+x-u, R^{**})]\end{aligned} \quad (55)$$

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

$$\begin{aligned}\tilde{V}_T(S+r-v, x) &= \min_{v' \in S+r-v+x} \mathbf{E} [V_T(S+r-v+x-v', R^{**})] \\ &= \mathbf{E} [V_T(S+r-v+x-v^*, R^{**})]\end{aligned} \quad (56)$$

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

$$\begin{aligned}\tilde{V}_T(S+r-u, x) - \tilde{V}_T(S+r-v, x) &= \mathbf{E} [V_T(S+r-v+x-u, R^{**})] \\ &\quad - \mathbf{E} [V_T(S+r-v+x-v^*, R^{**})]\end{aligned} \quad (57)$$

and here as well (50) follows by invoking the induction hypothesis once more, this time in state  $(S+r-v, x)$  as we note that any  $u \neq v$  in  $S+r$  is necessarily in  $S+r-v$ . This completes the proof of Theorem 1.