ON THE LIMITED UTILITY OF AUXILIARY INFORMATION IN THE LIST UPDATE PROBLEM

Micha Hofri[1] Dept. of Computer Science University of Houston

Hadas Shachnai[2] Dept. of Computer Science The Technion

April 20, 1997

Abstract

We consider the problem of dynamic reorganization of a linear list, where requests for the elements are generated randomly with *fixed* probabilities. The objective is to obtain the smallest expected cost *per access*. It has been shown that when no a-priori information is given on the reference probabilities, the *Counter Scheme (CS)* provides an optimal reorganization rule which applies to *all* possible distributions.

In this paper we examine strategies which may use partial *a priori* information on the access probabilities in the list reorganization process. Such useful knowledge may be either the correct relative order of a subset of the items, or the precise values of some of the probabilities.
For the first case we show that a slight modification on the original *CS* bests all *realizable* policies. However, it turns out that in the last case there exists no such universal optimal policy. For this model we also show that the obvious organization by the *Maximum Likelihood estimate* may be inferior to the *CS*, which totally *ignores* the auxiliary information.
We finally define the *Push forward Rule (PF)*, which is shown to be better than any rule which assumes no extra knowledge.

1 Introduction

The list update problem was first presented by McCabe [8] with the following layout: A fixed set of items is maintained as (an unsorted) linear list, as in a library, where books are positioned on a shelf, and each search for a book starts at the left end of the shelf and works to the right. The cost of accessing an item is determined by the length of this search. The list may be rearranged during a sequence of requests (possibly after each reference) so as to achieve a lower average access cost in subsequent requests.
We assume that each element may be accessed at any time with fixed probability. The goal is to arrange the elements 'correctly', i.e. in decreasing order of their access probabilities.
Previous works considered the situation where there is no *a-priori* information at all concerning the correct order of the elements. A comprehensive survey of many *permutation rules* suggested for the list management and their probabilistic analyses appears in [3].

Broadly speaking, such a rule tries to infer the best order of the list based on the sequence of references. The possible rearrangements are assumed to have negligible cost. This is reasonable for all the common policies. Recent works referred to the performance of reordering methods with respect to the online/offline measure, where *no initial assumptions* are made on the nature of the reference sequence to the list [2, 6, 9, 10].

[1]Department of Computer Science, The University of Houston, TX 77204-3475, USA.
e-mail: hofri@cs.uh.edu
[2]The Department of Computer Science, The Technion–IIT, Haifa 32000, Israel.
e-mail: hadas@cs.technion.ac.il.

Recent work [5] proves that under the above conditions the *Counter Scheme (CS)* produces the least expected cost of access *at any time*. The CS may move at most one record following each reference.

In the present paper we study a variant of the above model, where reordering rules may use some auxiliary information. certain Consider, for example, a merge of two lists, where each has already benefited from some reference history of its own. The obvious question is how should the complete structure be arranged to use this information to best advantage, and how it should be managed when subsequent requests are generated. A special case is where one of the lists is of length 1, in which the problem translates to efficient insertion of a single element, with *known* access probability, into a previously reorganized list. This is often the case in a list implementation of a dictionary, on which searches, insertions and deletions of records are permitted, and some of the new records are identified by their access probabilities. In a way, we propose to formulate an optimality statement – equivalent to the one in [5] – and suitable to this extension of the model.

Thus, we distinguish between degrees of partial *a priori* knowledge (about the reference probabilities) and discuss their contribution to the efficiency of the list reorganization process. A list model characterized by no initial information will be of the *Null Information (NI)* type, while any other model will be of *Partial Information (PI)*. Useful knowledge may be either the correct relative order of certain pairs of items in a given sub-list, or the *precise values* of some of the probabilities.

In Section 2 we discuss the notion of an optimal policy and define the class of policies which may be applied to the PI model.

We then discuss in section 3 the *symmetrical* model of partial information and define a policy which minimizes the average access cost at each reference for any choice of reference probabilities.

In section 4 we focus on the *asymmetric* model, for which we show the nonexistence of a globally optimal policy. We define the *MLE Rule (MLR)*, which is analogous to the *Counter Scheme*, which was shown to be the optimal strategy for the *NI* model. We show the surprising result that in spite of its reasonable use of *extra knowledge*, the *MLR* might be poorer than the *CS*, though both converge to the same asymptotic average cost. Finally we show how partial information on the values of the access probabilities *should* be used to improve the list order which results in the *NI* scenario.

We conclude in section 5 with discussion and a few related open problems.

2 Preliminaries and Notation

The structure we consider is a linear list of n records, $L = \{R_1, \ldots, R_n\}$. Each record R_i is uniquely identified by a key $K_i, 1 \le i \le n$.

Requests for the keys are drawn from a multinomial distribution driven by the reference probability vector (*rpv*): $\bar{p} = (p_1, \ldots, p_n)$. Thus, R_i may be accessed at any stage with a fixed probability p_i. This is the *independent reference model (irm)*.

Each reference requires a sequential search of the list. We define C, the cost of a single access, as the number of key comparisons made till the specified record is reached. Under the *irm*, with a fixed *rpv*, the average access cost to the list is minimized when the records are in the optimal static order: R_i precedes R_j whenever $p_i > p_j$. Getting there requires a complete knowledge of the *rpv*, or at least of the relative magnitude of the access probabilities. This knowledge is

assumed to be unavailable.
The initial arrangement is assumed to be randomly selected (with equal probability) out of all possible permutations, and then the list is constantly reorganized, with the aim of approaching the optimal ordering as the reference sequence grows longer.
Various performance measures were considered for that model, under a given policy H and an unknown *rpv* $\bar{p}$. The following will be used below:

1. The average access cost after the mth reference, $m \geq 0$, denoted below as $C_m(H \mid \bar{p})$
2. The expected access cost in the limiting state: $C(H \mid \bar{p}) = \lim_{m\to\infty} C_m(H \mid \bar{p})$
3. The rate of convergence of $C_m(H \mid \bar{p})$ to its limiting value, quantified by the *Overwork (OW)* measure [1], which is denoted by $OW(H \mid \bar{p})$.

Our definition of an optimal policy is natural: it is an admissible policy H^* that satisfies

$$C_m(H^* \mid \bar{p}) \leq C_m(H \mid \bar{p}), \quad \forall\ m, \bar{p} \text{ and admissible } H. \tag{1}$$

A policy is deemed admissible or realizable, if it

(i) does not use information about future references,

(ii) can be implemented without the benefit of information which is explicitly denied (such as the *rpv* values).

In an earlier paper [5] dealing exclusively with the *NI* case, we defined the set of symmetric or *key ignoring* policies $\mathcal{H}_{KI}$. A longer and more accurate name would be key-value ignoring policies. This simply means that throughout the reference history, the only criteria such a policy may use for distinguishing between the list records are their relative positions at every stage (including the initial one) and the history of requests for each.

Most of the scenarios in this paper have the list of records L split into two: L' has those records about which we have some information, and L'' – about which we are ignorant; Similarly to the NI case, we say a policy H is of type *l-KI*, for $2 \leq l \leq n$ when $|L''| = l$, and H uses at each stage a uniform, key-ignoring criterion for determining the relative order of R_i, R_j, $\forall\ R_i, R_j \in L''$.
We denote by $\mathcal{H}_{s.l}$ the set of *l-KI* policies where "*s*" stands for "symmetric".

3 Optimal Strategies for Order-PI Models

Assume our *a-priori* knowledge is limited to the correct relative order of a sublist L' of size l, say $R_1, \ldots, R_l$ (but not their actual *rpv* values). We call this *PI* model *local*, since the extra knowledge may only be used for rearranging the known sublist (as a separate structure). Any merge of L' with the "unknown" sublist L'' should respect this order. The policies for this model belong to the set $\mathcal{H}_{s.n-l}$.

Therefore, finding the optimal arrangement after a finite sequence of requests, may be done in two stages:

(i) Reordering 'optimally' the complete structure, under the assumption of no extra knowledge.

(ii) Using the auxiliary information to improve the list order.

We suggest the following policy, CS_2, which is such a *two stage* policy:
For any sequence of requests, the records of L'' are placed by their counters, i.e. if $c_i^{(m)}$ is the counter of $R_i \in L''$ after the mth reference, and it is the kth in size, then R_i will be positioned in the kth place. This leaves empty the positions that correspond to the L' records' counters. Next the vacant locations are given to L' in the correct order (ignoring the values of the counters).

Theorem 1: Within the set of admissible policies in $\mathcal{H}_{s.n-l}$, CS_2 minimizes the expected access cost to the list in each request.

Proof: Since the given information is irrelevant to the order of the records in L'', the CS ordering is best in the first stage. The irrelevancy claim follows from the following observation: Consider for brevity $l = 2$, let c_k stand for the counter value of R_k at some arbitrary time and pick $i, j > 2$. Then, using the multinomial distribution of the $\{c_k\}$ and Bayes rule we can compute probabilities such as $P(p_i < p_j | c_i < c_j)$, and we find

$$\frac{1}{2} < Pr(p_i < p_j | c_i < c_j) = Pr(p_i < p_j | c_i < c_j, p_1, p_2, \text{ and the order of } \; c_1, c_2 \text{ wrt } c_i \;\&\; c_j).$$

Since the additional information factors out and sums to one.
Then by an adversary type argument, it follows that any policy, which changes the location of any of the unknown records is not optimal (see [5]). Among the admissible policies which preserve the locations of the L'' elements, the one which places the sublist L' in correct relative order achieves the minimal expected access cost (any arrangement in which two of the known keys are 'incorrectly' located may be improved by an interchange). □

Another curious instance occurs when the entire *rpv* is known up to a permutation of the keys, i.e. *all* the access probabilities are given, but except p_1 they are *nameless*. In that scenario, after the mth reference, given that $c_1^{(m)} = c$ and any ordered counters vector $(c_2^{(m)}, \ldots, c_n^{(m)})$, the optimal position of R_1 may be determined by a direct computation when averaging over the $(n-1)!$ mappings of that vector to the indices $(2, \ldots, n)$. Hence we get an optimal realizable policy.

4 Reordering Methods for Value-PI Model

In this section we focus on a model, where l of the access *probabilities* $(1 \le l \le n-2)$ are initially *known*. Our first claim refers to the way in which partial knowledge of this sort disrupts our ability to specify a realizable optimal strategy. It suggests a clear distinction between the previous models, in which the *scalar* nature of the extra information allowed the definition of a *globally* optimal strategy, and the *numerical* information allowed by the present model which introduces an asymmetry that makes a crucial difference, as we now prove.
For simplicity, we formulate the following result for the case where $l = 1$. Nothing essential depends on this number, and with minor changes it holds for any $1 \le l \le n-2$.

Theorem 2: For any list of length $n > 2$, let $\frac{1}{n} < p_1 = p < \frac{1}{2}$ be known and fixed. Then no single policy within the class $\mathcal{H}_{s.1}$ minimizes the expected access cost at the mth request, $m > 1$, for all members of S_p:

$$S_p \equiv \{\bar{p} \mid p_1 = p\} \tag{2}$$

Proof: Our proof presents a partition of S_p with respect to the optimal policy. That is, distinct subsets in S_p will be shown to exhibit minimal access cost under different realizable policies. The key is that while any admissible policy must handle all *unknown* records symmetrically, it is allowed to single out R_1. Hence we define two natural policies.
H^*: This is the immediate extension of the CS to R_1. It can be implemented since p is known; for $i \in (2, \ldots, n)$ and whenever $c_1^{(m)} \neq m$, if

$$\frac{c_i^{(m)}(1-p)}{m - c_1^{(m)}} > p, \tag{3}$$

then place R_i ahead of R_1. When $c_1^{(m)} = m$, place R_1 first and the others in random order. Note that the left-hand side of the inequality is the efficient estimate of p_i.
H^{**}: The original *CS*.
Now consider the following *rpv*s (there is actually a single *rpv* in each of the sets we consider):

$$\bar{p}_a = (p, 1-p, 0, \ldots, 0). \tag{4}$$

By inspection, H^* is the optimal policy for this *rpv*, and

$$\bar{p}_b = (p, q, q, \ldots, q), \quad q = \frac{1-p}{n-1}, \tag{5}$$

where similarly H^{**} is the policy of choice for $\bar{p}_b$. The rest is a straightforward computation to show that H^* and H^{**} produce different permutations, and we omit the details. □

Our next result is an additional example for the above claim that auxiliary information may be of limited utility in the list reorganization process:
One approach to arguing for the optimality of *CS* (in the *NI* scenario) observes that it uses the counters to compute $\hat{p}_i$, the *Maximum Likelihood Estimates (MLE)* of the access probabilities. Under the *NI* model,

$$\hat{p}_i = \frac{c_i^{(m)}}{m}. \tag{6}$$

Similarly, when considering the present *PI* model, with l of the probabilities known, the *MLR* maintains the records ordered by the new estimates $\hat{p}_i$'s, where

$$\hat{p}_i = \begin{cases} p_i & 1 \leq i \leq l \\ \dfrac{c_i^{(m)}(1 - \sum_{j=1}^{l} p_j)}{m - \sum_{j=1}^{l} c_j^{(m)}} & l+1 \leq i \leq n, \quad \sum_{j=1}^{l} c_j^{(m)} < m \\ 0 & l+1 \leq i \leq n, \quad \sum_{j=1}^{l} c_j^{(m)} = m \end{cases}$$

A rather surprising distinction between the *MLR* and the *CS* may now be shown:
Let σ_m the list arrangement after the mth reference, and $\sigma_m(i)$ the location of R_i in it. Then whenever $p_i > p_j, 1 \leq i, j \leq n$,

$$Pr_{CS}[\sigma_m(i) < \sigma_m(j)] \longrightarrow 1, \tag{7}$$

monotonically, for $m \to \infty$, while for any $1 \leq j \leq l$ and $l+1 \leq i \leq n$ the value of

$$Pr_{MLR}[\sigma_m(i) < \sigma_m(j)], \tag{8}$$

while it converges to 1 as well, is *not necessarily* monotone in m. It will be seen below that this is equivalent to the statement that the cost under *CS* can be strictly smaller than under the *MLR*. We demonstrate this for $l = 1$ and a specific class of *rpvs*.

¿From a statistical point of view the above $\hat{p}_i$ represents our best estimate of the *rpv*, and hence we would expect that using it in the way the *MLR* does should lead to an optimal re-arrangement of the records. This may well be often the case, but we can show the surprising result that there are exceptions.

As both *CS* and *MLR* converge to the optimal ordering, a criterion for comparing their transient behaviour may be the *overwork (OW)* measure. As defined in [1], the *overwork* of a policy H is the area between the cost curve $C_m(H \mid \bar{p})$ and its asymptote $C(H \mid \bar{p})$.

In the following observation, we assume the known probability is $p_n = p$, to simplify the notation.

Observation 1: For some *rpvs* $\bar{p} = (p_1, p_2, \ldots, p_n)$ satisfying $p_n = min_{1 \leq i \leq n} p_i = p$,

$$OW(MLR \mid \bar{p}) > OW(CS \mid \bar{p}). \tag{9}$$

Proof: Assume a temporary renumbering of the records so that their probabilities, now denoted by $p_{(i)}$, satisfy $p_{(1)} \geq p_{(2)} \ldots \geq p_{(n)} = p$. With this notation, where OPT stands for the optimal order,

$$C_m(OPT \mid \bar{p}) = C(OPT \mid \bar{p}) = \sum_{i=1}^{n} i p_{(i)}.$$

For $H \in \{MLR, CS\}$,

$$\begin{aligned} C_m(H \mid \bar{p}) &= 1 + \sum_{i=1}^{n} p_{(i)} \sum_{j \neq i} Pr_H(\sigma_m(j) < \sigma_m(i)) \\ &= 1 + \sum_{i=1}^{n} \sum_{j=i+1}^{n} p_{(i)} Pr_H(\sigma_m(j) < \sigma_m(i)) + p_{(j)}(1 - Pr_H(\sigma_m(j) < \sigma_m(i))) \\ &\equiv C(OPT \mid \bar{p}) + OW_m(H \mid \bar{p}), \end{aligned}$$

where $OW_m(H \mid \bar{p})$ is the contribution of step m to the overwork measure, such that

$$OW(H \mid \bar{p}) \equiv \sum_{m \geq 0} OW_m(H \mid \bar{p}). \tag{10}$$

¿From the definition of $C_m(H \mid \bar{p})$ we see that

$$OW_m(H \mid \bar{p}) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} (p_{(i)} - p_{(j)}) Pr_H(\sigma_m(j) < \sigma_m(i)).$$

Let us introduce further notation: for a fixed *rpv* $\bar{p}$ and any pair of indices $1 \leq i < j \leq n$ (which implies $p_{(i)} \geq p_{(j)}$), define

$$OW_{ij}(H \mid \bar{p}) \equiv (p_{(i)} - p_{(j)}) \sum_{m \geq 0} Pr_H(\sigma_m(j) < \sigma_m(i)). \tag{11}$$

Then,

$$OW(H \mid \bar{p}) = \sum_{1 \le i < j \le n} OW_{ij}(H \mid \bar{p}). \tag{12}$$

Now, for all $1 \le i < n$, the *MLR* would assign $\sigma_m(i) < \sigma_m(n)$, when $c_n^{(m)}$ satisfies $\lceil (m - c_n^{(m)})p/(1-p) \rceil \le c_i^{(m)}$ for all $1 \le m$ (to simplify the argumentation, an irrational value for p can be assigned, and then the ratio $(m - c_n^{(m)})p/(1-p)$ is never an integer). Hence

$$Pr_{MLR}[\sigma_m(i) < \sigma_m(n)] < 1 - (1 - p_i)^m, \tag{13}$$

since the right-hand side is the probability that R_i is at all referenced. Thus, from equation (11)

$$OW_{in}(MLR \mid \bar{p}) > (p_{(i)} - p) \sum_{m \ge 0} (1 - p_{(i)})^m = \frac{p_{(i)} - p}{p_{(i)}}. \tag{14}$$

Now, it is shown in [4], that

$$OW_{ij}(CS \mid \bar{p}) \le \frac{p_{(i)} + p_{(j)}}{2(p_{(i)} - p_{(j)})} \quad \forall \quad 1 \le i, j \le n, \text{ where } p_{(i)} > p_{(j)}. \tag{15}$$

It only remains now to observe that there are values of $p_{(i)} > p$, such that the last two inequalities allow us to conclude that

$$OW_{in}(MLR \mid \bar{p}) > OW_{in}(CS \mid \bar{p}). \tag{16}$$

In order to show that such *rpv*s exist, it suffices to choose a particular case. So here is a particularly simple one: let $p_i = rp$, for all $1 \le i \le n-1$, and substitute this into the inequality

$$\text{Right-hand side of (14)} \ge \text{Right-hand side of (15) with} j = n, \tag{17}$$

and the resulting quadratic (in r) is satisfied for any $r > (5 + \sqrt{17})/2 \approx 4.56$, yielding the inequality (16). Since for all pairs not involving R_n,

$$OW_{ij}(MLR \mid \bar{p}) = OW_{ij}(CS \mid \bar{p}), \tag{18}$$

we get the inequality (9). □

The last result implies, that there is at least one value of m, where $C_m(MLR|\bar{p}) > C_m(CS|\bar{p})$. Hence the surprising fact that in some cases, although p_n is known, we could do better using the *estimate* $\hat{p}_n$ for positioning R_n in the list, rather than the actual value of p_n.

Obviously, this is unexpected from a statistical point of view. However, it is not difficult to account for, because of our particular choice for the example: when m is small, and many of the counters are still at 0, MLR – knowing $p_n > 0$ – would put R_n, although it has the *smallest* access probability in this example, still in front of all the records with zero counters. The CS, however, since c_n is very likely to be then 0 as well, would leave it at a random place in the collection of all these records, at the back of the list, and obtain a lower expected cost. As m increases more and more records would have their estimated $\hat{p}_i$ exceed p_n and placed by the MLR ahead, of R_n, where they should be. This corresponds to the fact that both policies converge to the optimal order.

We are going now to change our tone. So far we have shown that the presence of partial information on precise values of some of the access probabilities raises problems in choosing a reorganization rule which is globally optimal. Yet a proper use of the knowledge of such

values, in conjunction with the CS, can *improve* the ordering of the records, as suggested by the following result. This is a direct generalization of the observation that if we knew p_1 to exceed one half, we would place R_1 first, since that value must be the largest. Similarly, at most $\lfloor (1-p)/p \rfloor$ records can have access probabilities that exceed p.
Let $V = (v_1, \ldots, v_n)$ be a vector of n components, where $v_i \in N$, and

$$V_{-i} \equiv (v_1, \ldots v_{i-1}, v_{i+1}, \ldots v_n), \tag{19}$$

$$\bar{V} = (v_{(1)}, \ldots, \ldots, v_{(n)}), \quad \text{such that } v_{(i)} \geq v_{(j)} \text{ whenever } i < j. \tag{20}$$

$\bar{V}_{-i}$ denotes the sorted vector where v_i (not $v_{(i)}$) is omitted.
Assume the records are so numbered that $1 \geq p_{(1)} \geq \ldots p_{(n)} \geq 0$.

Lemma 1: Let the vector of n counter values following the mth reference when *reordered according to the above renumbering* be denoted by $\bar{C}^{(m)}$. The function

$$f(k) = \sum_{i=1}^{n} p_{(i)} Pr_{CS}(\sigma_m(i) = k \mid \bar{C}^{(m)}), \quad \forall \; 1 \leq k \leq n \tag{21}$$

is monotone non-increasing in k.

Proof: This is a special case of a more general property which is immediate to prove: if the sequence a_i of real numbers is monotonically non-increasing, and B is a matrix with elements that satisfy $i < j \Longrightarrow b_{ij} \geq b_{ji}$, then $\sum_i a_i \sum_j (b_{ij} - b_{ji}) \geq 0$. □

Assume as before that $p_1 \equiv p$ is known. Let $c_1 \equiv c_1^{(m)}$ denote the counter value of R_1 after m requests. We use $\hat{\sigma}_m$ to describe the relative order of $R_2, \ldots, R_n$ in the $n-1$ locations they occupy. From Lemma 1 we have

Corollary 3: For any ordered frequency vector $\bar{C}_{-1}^{(m)}$, the function

$$f_{-1}(k \mid \bar{C}_{-1}^{(m)}) = \sum_{i=2}^{n} p_i \Pr_{CS}(\hat{\sigma}_m(i) = k \mid \bar{C}_{-1}^{(m)}), \tag{22}$$

is monotonically non-increasing in k, $\forall \; 1 \leq k \leq n-1$.

We use $\sigma_m^{CS}(i)$ to denote the position of R_i under the CS after the mth reference to the list.
The *Push Forward (PF) Rule* restricts the optimal locations for any element R_i, with known access probability. Specifically, if $p_1, p_2, \ldots, p_k$ are known and numbered in nonincreasing order, define

$$1 \leq \lfloor (1 - \sum_{j=1}^{k} p_j)/p_i \rfloor \equiv l_i^* < n, \quad 1 \leq i \leq k, \tag{23}$$

then for any sequence of references, the list is first ordered by the CS, and then, for $1 \leq i \leq k$

$$\sigma_m^{PF}(i) = min(\sigma_m^{CS}(i), i + l_i^*) \tag{24}$$

Theorem 4:

$$C_m(PF \mid \bar{p}) < C_m(CS \mid \bar{p}). \tag{25}$$

for all $m \geq 1$.

Proof: For simplicity, we handle the case of $k=1$. The proof can be easily generalized. We show below that if p_1 is *known*, then it never pays to put R_1 beyond position $l^* \equiv 1 + l_1^*$. It is sufficient to show, that for all possible $\bar{C}^{(m)}$, the counter-values vector following m references

$$C_m(PF \mid \bar{p}, \bar{C}^{(m)}) \leq C_m(CS \mid \bar{p}, \bar{C}^{(m)}). \tag{26}$$

We first show that for any given $\bar{C}^{(m)}$ and $k > l^*$, if the policies $H \in \mathcal{H}_{s.n-1}$ and CS differ by one in the position they assign for R_1,

$$\sigma_m^{CS}(1) = k \quad \text{and} \quad \sigma_m^{H}(1) = k-1 \tag{27}$$

with the same relative order of $R_2, \ldots, R_n$, then

$$C_m(H \mid \bar{p}, \bar{C}^{(m)}) < C_m(CS \mid \bar{p}, \bar{C}^{(m)}) \tag{28}$$

By definition of the cost function we get with routine manipulations

$$C_m(CS \mid \bar{p}, \bar{C}^{(m)}) - C_m(H \mid \bar{p}, \bar{C}^{(m)}) = p - f_{-1}(k-1 \mid \bar{C}_{-1}^{(m)}) \tag{29}$$

where $f_{-1}(k \mid \bar{C}_{-1}^{(m)})$ is defined in equation (22).
The proof is by way of contradiction. Assume,

$$f_{-1}(k-1 \mid \bar{C}_{-1}^{(m)}) \geq p. \tag{30}$$

Then, by corollary 3 for all l in the range $1 \leq l \leq k-2$ also

$$f_{-1}(l \mid \bar{C}_{-1}^{(m)}) \geq p. \tag{31}$$

Now, using the definition in equation (22)

$$\begin{aligned}\sum_{l=1}^{k-1} f_{-1}(l \mid \bar{C}_{-1}^{(m)}) &= \sum_{l=1}^{k-1}\sum_{i=2}^{n} p_i \mathrm{Pr}_{CS}(\hat{\sigma}_m(i) = l \mid \bar{C}_{-1}^{(m)}) \\ &= \sum_{i=2}^{n} p_i \sum_{l=1}^{k-1} \mathrm{Pr}_{CS}(\hat{\sigma}_m(i) = l \mid \bar{C}_{-1}^{(m)}) \leq \sum_{i=2}^{n} p_i \sum_{l=1}^{n-1} \mathrm{Pr}_{CS}(\hat{\sigma}_m(i) = l \mid \bar{C}_{-1}^{(m)}) = 1-p.\end{aligned}$$

Hence

$$\sum_{l=1}^{k-1} f_{-1}(l \mid \bar{C}_{-1}^{(m)}) \leq 1-p. \tag{32}$$

Using equation (31) and $1 < \lceil (1-p)/p \rceil = l^* < n$, we find

$$\sum_{l=1}^{k-1} f_{-1}(l \mid \bar{C}_{-1}^{(m)}) \geq (k-1)p. \tag{33}$$

But as $1 - p < l^* p < (k-1)p$, we also get

$$\sum_{l=1}^{k-1} f_{-1}(l \mid \bar{C}_{-1}^{(m)}) > 1-p, \tag{34}$$

which contradicts relation (32).
Hence the assumption (30) is false, the left-hand side of (29) is positive, and

$$C_m(H \mid \bar{p}, \bar{C}^{(m)}) < C_m(CS \mid \bar{p}, \bar{C}^{(m)}). \tag{35}$$

Since this has been shown to hold for all $l^* < k \leq n$, the *PF Rule*, which is the limit (in the way it positions R_1) of all the above policies H satisfies, for some $\bar{C}^{(m)}$,

$$C_m(PF \mid \bar{p}, \bar{C}^{(m)}) < C_m(CS \mid \bar{p}, \bar{C}^{(m)}) \tag{36}$$

and

$$C_m(PF \mid \bar{p}) < C_m(CS \mid \bar{p}). \tag{37}$$

This proves relation (25) with a $\geq$ inequality. The sharpness follows from our ability to exhibit vectors $\bar{C}^{(m)}$ of non-zero probability which lead to PF actually modifying the CS order. □

5 Concluding Remarks

We have studied a variation of the classical dynamic list model and have shown that frequency counts may be of limited merit when coupled with partial information on the precise values of access probabilities. However, they are still the best choice in the symmetric, or local *PI* situations.

Though Theorem 2 considers a model in which the number of records is fixed, it may be easily generalized for a list of varying length:

A fortiori, there is no optimal rule for placing a new record R_{n+1}, with the known access probability p_{n+1}, in a list when the ratios between old access probabilities are preserved, i.e. if R_{n+1} is inserted with the mth reference, then

$$\frac{{p_i}^{(m)}}{{p_j}^{(m)}} = \frac{{p_i}^{(m-1)}}{{p_j}^{(m-1)}} \quad \forall \quad 1 \leq i, j \leq n \tag{38}$$

(We assume $\sum_{i=1}^{n+1} {p_i}^{(m)} = 1, \forall \ m \geq 1$).

This holds when trying to minimize the expected access cost at each reference. Obviously, the difficulty only arises for the transient analysis: the placement chosen for R_{n+1} in the list, following its insertion, will not affect the asymptotic optimality of the rule applied, as the other access probabilities are eventually estimated well enough to construct the best ordering. But in a dynamic environment, limiting optimality has only a limited attration.

We conclude with the comment that a result such as presented in Theorem 2 may not hold for other models of self-organizing sequential search. An example is the dynamic path tables described in [11]:

Each path i has a fixed but unknown failure probability p_i, independently of all other path failures. When trying to route a new message, the table is scanned from the top, till the first success (or exhaustion of all paths). Then a new permutation of the paths is chosen. Clearly, knowing any sub-set of these failure probabilities will not reduce in advance the range of locations among which those paths are to be positioned, due to the mutual *independence* between the p_i's (in particular, we have no constraint on their sum).

References

[1] Bitner, J., "Heuristics that Dynamically Organize Data Structures", *SIAM J. Comput.*, **8**,#1, pp. 82-110, 1979.

[2] Bentley J. L., McGeoch C.C, "Amortized Analyses of Self-Organizing Sequential Search Heuristics", *Commun. ACM*, **28**, #4, pp. 404-411, 1985.

[3] Hester J.H., Hirschberg D.S., "Self-Organizing Linear Search", *ACM Comput. Surv.*, **17**, #3, pp. 295-312, 1985.

[4] Hofri M., Shachnai H., "Self-Organizing Lists and Independent References - a Statistical Synergy", *Jour. of Alg.*, **12**, pp. 533-555, 1991.

[5] Hofri M., Shachnai H., "On the Optimality of Counter Scheme for Dynamic Linear Lists", *Inf. Process. Lett.*, **37**, pp. 175-179, 1991.

[6] Irani S., Reingold N., Westbrook J., Sleator D., "Randomized Competitive Algorithms for the List Update Problem", *private communication*,

[7] Lam K., Siu M.K., Yu C.T., "A Generalized Counter Scheme", *Theor. Comput. Sci.*, **16**, #3, pp. 271-278, 1981.

[8] McCabe J., "On Serial Files with Relocatable Records", *Operations Research*, **13**, pp. 609-618, 1965.

[9] Reingold N., Westbrook J., "Optimal Off-line Algorithms for the List Update Problem", *Technical Report YALEU/DCS/TR-805* Yale University, June 1990.

[10] Sleator D.D., Tarjan R.E., "Amortized Efficiency of List Update and Paging Rules", *Commun. ACM* **28**, #2, pp. 202-208, 1985.

[11] Topkis D. M., "Reordering Heuristics for Routing in Communication Network", *J. of Applied Prob.*, **23**, pp. 130-143, 1986.