ON THE OPTIMALITY OF THE COUNTER SCHEME FOR DYNAMIC LINEAR LISTS

Micha Hofri† and Hadas Shachnai

Computer Science Department
The Technion - Israel Institute of Technology, Haifa 32000, Israel
September 1990

ABSTRACT

We consider policies that manage fixed-size dynamic linear lists, when the references follow the *independent reference model*. We define the *counter scheme*, a policy that keeps the records sorted by their access frequencies, and prove that among all deterministic policies it produces the least expected cost of access, at any time.

1. Introduction

We consider a linear list of n records, $\{R_i\}_{i=1}^{n}$. An access to R_i requires a sequential search of the list starting at the header, till R_i is encountered. The *cost* of a single access is defined to be the number of keys examined in the search.

Assumption: The reference history is a series of independent multinomial trials, with fixed but unknown reference-probability vector (*rpv*) $\boldsymbol{p} = (p_1, \cdots, p_n)$. This is the *independent reference model (irm).*

The problem of minimizing the expected access cost, using dynamic reorganization of the list, has been widely studied (see Hofri and Shachnai 1988, and further references there). Most of the suggested organization rules incur no storage overhead, and are called *memory free*; typical representatives are *Move To the Front* (*MTF*), which places an accessed record at the head of the list, leaving the other elements untouched, and the *Transposition Rule* (*TR*), which advances the referenced record one step ahead by an interchange with its preceding neighbor.

Rules that use additional storage are naturally less appealing compared with the previous methods. However, their relative efficiency in the list reorganization process might compensate for their space complexity. We focus on *Counter Scheme* (*CS*), which handles the list in the following manner:

A frequency counter c_i stores the number of accesses to the record R_i, $1 \le i \le n$, throughout the reference history. The list is maintained sorted, in nonincreasing order of the counter values.

When asymptotic (expected) cost is considered, the *CS* achieves the optimum; in this sense it dominates all other common permutation rules. It is also known to have advantages in the finite horizon case, when the average access cost following a *finite* sequence of requests to the list is considered. This was shown by Lam *et al.* (1981) when analyzing their *Generalized Counter Scheme*, a special instance of which is the above *CS*. They proved that *CS* is better than any other possible *counter based* method.

---

† Currently at the Department of Computer Science, University of Houston, Houston, Tx 77402-3475, USA.

There are many other possible policies. Generally, a realizable (or admissible) policy is any policy that (a) has no advance knowledge about $\boldsymbol{p}$, and (b) does not know the future references. The reorganization may depend on the order at which records were referenced, on their location when referenced (and *TR* is a special case of this), on the number of times a record was moved, on the highest (lowest) position it has so far occupied, on all of the above *and* the counters... A noteworthy fact is that an optimal policy *does* exist. For example, it is known that among all memory-free policies there is none which is optimal when no information is available about $\boldsymbol{p}$. Our purpose is to strengthen the result of Lam *et al.* and prove that *CS* is optimal among **all** realizable policies with respect to the average cost at the mth request, for **any** $m \geq 1$.

From a statistical point of view this is hardly surprising: the optimal order only depends on the ranking of the probabilities $\{p_i\}$; the counters $\{c_i\}$ are known to be sufficient statistics for the $\{p_i\}$. A-priori they should then suffice to compute an optimal policy.

2. Proof of Optimality

Assume the initial state of the list is random, with equal probability for each of the $n!$ orderings. The arrangement of the records after the mth request, also known as "at time m", is represented as

$$\sigma_m = \begin{pmatrix} 1 & 2 & \cdots n \\ \sigma_m(1) & \sigma_m(2) & \cdots \sigma_m(n) \end{pmatrix} \qquad (1)$$

with $\sigma_m(i)$ = the position of R_i. We shall use below σ_m also when interpreted as a permutation operator, with the usual definition of multiplication as successive application (denoted by the symbol $\circ$).

We define a history of references at time m, under the policy H, as the vector

$$I^{(m)} \equiv (i_1, \cdots, i_m), \qquad (2)$$

where i_k denotes the record accessed at the kth request. $I^{(m)}$ is called the *reference history vector (rhv)*.

The following notation is basic to our proof method:

$$\bar{\sigma}_m = \begin{pmatrix} 1 & \cdots & n \\ \sigma_m(\sigma_0^{-1}(1)) & \cdots & \sigma_m(\sigma_0^{-1}(n)) \end{pmatrix}. \qquad (3)$$

Here, $\bar{\sigma}_m$ denotes the *canonical ordering* of the list after the mth request: given an initial state σ_0, each record is identified by its original position in the list. (We could formulate this as a transformation on the record *name space*). For any initial order, $\bar{\sigma}_0$ is the identity permutation, and $\bar{\sigma}_m$ describes the list-order at time m in terms of the initial positions of the records.

The *rhv* $I^{(m)}$, when expressed in terms of the canonical representation produces $\bar{I}^{(m)}$, the *canonical history vector* (*chv*):

$$\bar{I}^{(m)} \equiv (\bar{i}_1, \cdots, \bar{i}_m), \qquad \bar{i}_k = \sigma_0(i_k). \qquad (4)$$

Finally, let $\bar{C}^{(m)} = (\bar{c}_1^{(m)}, \cdots, \bar{c}_n^{(m)})$ be the canonical frequency vector (*cfv*) accumulated during a sequence of m references, where $\bar{c}_i^{(m)}$ is the counter of the record positioned ith in the initial order. The notation $\Pr_H(\sigma_m | I^{(m)}, \sigma_0)$ is defined as the probability that a policy H will carry the initial order σ_0, under the reference history $I^{(m)}$ (implicitly generated by an *irm* source − fixed but unknown) into the final state σ_m.

We introduce now two classes of policies:
H_D stands for the class of *deterministic* permutation rules: for a given initial ordering σ_0 and a reference history $I^{(m)}$, the outcome σ_m is defined by H uniquely, for all $m \geq 1$.

H_{KI} denotes the class of *key-ignoring* policies. While there appears to be no difficulty with the intuitive notion, a precise definition of H_{KI} requires some care. A policy H will be said to be in H_{KI} when it satisfies the following constraint: Consider a pair of initial orderings $\sigma_0^{(1)}, \sigma_0^{(2)}$ and the permutation $g = \sigma_0^{(2)^{-1}} \circ \sigma_0^{(1)}$. Then for every history, expressed by the canonical vector $\overline{I}^{(m)}$, we find

$$\Pr_H(\sigma_m \circ g \mid \overline{I}^{(m)}, \sigma_0^{(1)}) \;=\; \Pr_H(\sigma_m \mid \overline{I}^{(m)}, \sigma_0^{(2)})\,. \tag{5}$$

When H is deterministic this merely says that the effect of $\overline{I}^{(m)}$ is invariant with respect to the names chosen for the records[1]. Hence policies in H_{KI} are adequate. Alternatively, one may be easily convinced, by adversary-type arguments, that any policy which is not *key-ignoring*, would not be optimal under the *irm* assumptions.

Let $H_{DK} = H_D \cap H_{KI}$. The next Lemma shows that we may restrict our attention to H_{DK}:

Lemma 1: Within the class of H_{KI}, there exists a policy $H \in H_{DK}$ which minimizes the average access cost at time m, $m \geq 1$.

We leave out the proof; it uses induction on m to show that any non-deterministic rule in H_{KI} cannot do better than the best strategy in H_{DK}.

Consider two initial orderings $\sigma_0^{(1)}, \sigma_0^{(2)}$ which differ only by the interchange of two records R_i, R_j:

$$\begin{aligned} &\sigma_0^{(1)}(i) = k\,, \quad \sigma_0^{(1)}(j) = l \qquad k < l \\ &\sigma_0^{(2)}(i) = l\,, \quad \sigma_0^{(2)}(j) = k \qquad \sigma_0^{(1)}(s) = \sigma_0^{(2)}(s) \qquad 1 \leq s \leq n, \quad s \neq i, j. \end{aligned} \tag{6-0}$$

Two observations about this notation, formulated as lemmas, provide the tools for the main result.

Lemma 2: For all $H \in H_{DK}$, the permutations $\sigma_m^{(1)}, \sigma_m^{(2)}$ produced using H with the *chv* $\overline{I}^{(m)}$ and the initial orders $\sigma_0^{(1)}, \sigma_0^{(2)}$ respectively, satisfy

$$\sigma_m^{(1)}(i) = \sigma_m^{(2)}(j)\,, \quad \sigma_m^{(1)}(j) = \sigma_m^{(2)}(i)\,; \quad \sigma_m^{(1)}(s) = \sigma_m^{(2)}(s)\,, \quad \forall\ s \neq i, j\ . \tag{6-m}$$

The proof is immediate: since $H \in H_{DK}$, equation (5) yields $\sigma_m^{(1)} = \sigma_m^{(2)} \circ \sigma_0^{(2)^{-1}} \circ \sigma_0^{(1)}$. Using the relation (6-0) and performing the multiplication leads to the relation (6-m). □

Let $\Pr_H(\sigma_m(i) < \sigma_m(j) \mid \overline{I}^{(m)}, \sigma_0)$ denote the probability that R_i precedes R_j in the list after the m-th reference, when using the policy H, given the occurrence of the *chv* $\overline{I}^{(m)}$, and that the initial state was σ_0. Similarly, $\Pr_H(A, S)$ is used to denote the joint probability of the events A and S. Lemma 3 rephrases Lemma 2 in a form convenient for our use:

Lemma 3: For any canonical reference history vector $\overline{I}^{(m)}$ and $H \in H_{DK}$, with $\sigma_0^{(1)}$ and $\sigma_0^{(2)}$ defined as above,

$$\Pr_H(\sigma_m(i) < \sigma_m(j) \mid \overline{I}^{(m)}, \sigma_0^{(1)}) = \Pr_H(\sigma_m(j) < \sigma_m(i) \mid \overline{I}^{(m)}, \sigma_0^{(2)})\ . \tag{7}$$

Now, denote by U the event, that the initial state is either $\sigma_0^{(1)}$ or $\sigma_0^{(2)}$, and the history of references produces the *chv* $\overline{I}^{(m)}$. The next Lemma states explicitly that CS does a better job at approximating the "correct" order of the records than any other policy for any possible reference

---

[1]In the general case, when H is not necessarily deterministic, the last requirement means that the sequence $\{\sigma_m\}$ has to be measurable with respect to the increasing σ-algebra generated by the sequence $\{\overline{I}^{(m)}\}$, which is key-*ignorant*.

history.

Lemma 4: For an arbitrary policy $H \in H_{DK}$, and any pair of records R_i, R_j, $1 \le i \ne j \le n$, with respective access probabilities p_i, p_j, the following implication holds:

$$p_i > p_j \implies \Pr_{CS}(\sigma_m(i) < \sigma_m(j), U) \ \ge\ \Pr_H(\sigma_m(i) < \sigma_m(j), U)\,. \tag{8}$$

Proof: Define $x \equiv \Pr_H(\sigma_m(i) < \sigma_m(j) \mid \bar{I}^{(m)}, \sigma_0^{(1)})$. Then, by Lemmas 2 and 3, the corresponding probability $\Pr_H(\sigma_m(i) < \sigma_m(j) \mid \bar{I}^{(m)}, \sigma_0^{(2)})$ equals $1-x$. Observe that in general x may depend on arbitrary features of $\bar{I}^{(m)}$, and can be any value in $[0,1]$, but that in the case $H = CS$, $x \in \{0,1\}$. Then, using the multinomial coefficient $\binom{m}{\bar{C}^{(m)}}$, where $\bar{C}^{(m)}$ is the *cfv* resulting from $\bar{I}^{(m)}$,

$$\Pr_H(\sigma_m(i) < \sigma_m(j), U) = \Pr_H(\sigma_m(i) < \sigma_m(j)\ , \bar{I}^{(m)}, \sigma_0^{(1)}) + \Pr_H(\sigma_m(j) < \sigma_m(i)\ , \bar{I}^{(m)}, \sigma_0^{(2)})$$

$$= \Pr_H(\sigma_m(i) < \sigma_m(j) \mid \bar{I}^{(m)}, \sigma_0^{(1)}) \cdot \Pr(\bar{I}^{(m)}, \sigma_0^{(1)}) + \Pr_H(\sigma_m(i) < \sigma_m(j) \mid \bar{I}^{(m)}, \sigma_0^{(2)}) \cdot \Pr(\bar{I}^{(m)}, \sigma_0^{(2)})$$

$$= x \cdot \frac{1}{n!}\binom{m}{\bar{C}^{(m)}}\Big(\prod_{r \ne i,\, j} p_r^{c_r}\Big) p_i^{\bar{c}_k} p_j^{\bar{c}_l} + (1-x) \cdot \frac{1}{n!}\binom{m}{\bar{C}^{(m)}}\Big(\prod_{r \ne i,\, j} p_r^{c_r}\Big) p_i^{\bar{c}_l} p_j^{\bar{c}_k} \tag{9}$$

$$= \frac{1}{n!}\binom{m}{\bar{C}^{(m)}}\Big(\prod_{r \ne i,\, j} p_r^{c_r}\Big)(x \cdot p_i^{\bar{c}_k} p_j^{\bar{c}_l} + (1-x) \cdot p_i^{\bar{c}_l} p_j^{\bar{c}_k})\ .$$

Similarly, for $H = CS$,

$$\Pr_H(\sigma_m(i) < \sigma_m(j), U) = \begin{cases} \frac{1}{n!}\binom{m}{\bar{C}^{(m)}}\Big(\prod_{r \ne i,\, j} p_r^{c_r}\Big) p_i^{\bar{c}_k} p_j^{\bar{c}_l} & \bar{c}_k > \bar{c}_l \\ \frac{1}{n!}\binom{m}{\bar{C}^{(m)}}\Big(\prod_{r \ne i,\, j} p_r^{c_r}\Big) p_i^{\bar{c}_l} p_j^{\bar{c}_k} & \bar{c}_l \ge \bar{c}_k \end{cases} \tag{10}$$

When comparing these two probabilities all common factors cancel out, leaving just p_i and p_j. Now, if $\bar{c}_k > \bar{c}_l$, then

$$p_i^{\bar{c}_k}\ p_j^{\bar{c}_l} = x p_i^{\bar{c}_k}\ p_j^{\bar{c}_l} + (1-x) p_i^{\bar{c}_k}\ p_j^{\bar{c}_l} \quad > \quad x p_i^{\bar{c}_k}\ p_j^{\bar{c}_l} + (1-x)\ p_i^{\bar{c}_l}\ p_j^{\bar{c}_k}.$$

and symmetrically, in the case where $\bar{c}_l \ge \bar{c}_k$,

$$p_i^{\bar{c}_l}\ p_j^{\bar{c}_k} = x p_i^{\bar{c}_l}\ p_j^{\bar{c}_k} + (1-x) p_i^{\bar{c}_l}\ p_j^{\bar{c}_k} \quad > \quad x p_i^{\bar{c}_l}\ p_j^{\bar{c}_k} + (1-x)\ p_i^{\bar{c}_k}\ p_j^{\bar{c}_l},$$

and the inequality in the Lemma follows. □

Taking the marginal distribution in relation (8), by summing out U, we have

Corollary 5:

$$p_i > p_j \implies \Pr_{CS}(\sigma_m(i) < \sigma_m(j)) \ \ge\ \Pr_H(\sigma_m(i) < \sigma_m(j))\,. \qquad \square$$

Let $C_m(H|\boldsymbol{p})$denote the expected access cost to the list after the mth request, using the policy H, where the expectation is evaluated over all $m+1$-long histories and $n!$ initial orders. Our main result is

Theorem: For the linear-list model described in Section 1, under any admissible policy H,

$$C_m(CS|\boldsymbol{p}) \le C_m(H|\boldsymbol{p})$$

for all $m \ge 1$.

Proof: Use the above discussion to limit consideration to $H \in H_{DK}$. Then, without loss of generality, assume that $p_i \ge p_j$ whenever $i < j$. Splitting the cost to a sum on record pairs we find,

$$
\begin{aligned}
C_m(H|\boldsymbol{p}) &= \sum_{1\le i<j\le n} (p_i \, \Pr_H(\sigma_m(j) < \sigma_m(i)) + p_j \, \Pr_H(\sigma_m(i) < \sigma_m(j)) \\
&= 1 + \sum_{1\le i<j\le n} (p_j - p_i) \Pr_H(\sigma_m(i) < \sigma_m(j)) \\
&\ge 1 + \sum_{1\le i<j\le n} (p_j - p_i) \Pr_{CS}(\sigma_m(i) < \sigma_m(j)) \quad = \quad C_m(CS|\boldsymbol{p}) \, . \qquad \square
\end{aligned}
$$

3. Further Remarks

We have shown that *CS* is the optimal reorganization method not only in the limiting sense, but for any finite sequence of requests.

To avoid the allocation of huge counter fields, *CS* may be replaced by the *Limited Counters Scheme (LCS)* (Hofri and Shachnai, 1988). This 'truncated' version of *CS* reduces significantly its storage requirements while still being very effective. It would be of interest to examine the classes of policies which can still do better than the various versions of *LCS*.

We comment that the optimality of *CS* holds under the following assumptions on the model :

(i) The set of records in the list remains fixed over time.

(ii) No initial information on the *rpv*.

(iii) Independent and time-homogeneous reference probabilities.

Permitting insertions and deletions, or having some a-priori knowledge of any subset of the access probabilities may lead to new conclusions concerning the existence of an optimal policy and its thus-implied characteristics. We are currently pursuing some of these problems.

Relaxing the independence assumption has not been considered in previous work. We believe that for certain models of dependent references, the optimality of *CS* still holds, albeit with a different character. This is certainly the case when the components of $\boldsymbol{p}$ are time-varying, but without changing their ranking. For a different one, assume a reference model which follows a first-order Markov chain, i.e. p_{ij} is the conditional probability of accessing R_j after a reference to R_i, $1 \le i, j \le n$. If none of those transition probabilities is known in advance, and the same cost structure holds (where key-comparisons carry a price tag but record shuffles do not), consider the following reorganization scheme :

Each of the records is associated with a frequency vector $\boldsymbol{C}_i$, where $C_{i,j}$ counts the number of accesses to R_j immediately following a request to R_i. Then a reference to R_i (preceded by a search for R_k) would result with an increment of the appropriate counter ($C_{k,i}$) and a new permutation of the list – in descending order of the counters $C_{i,j}$, $1 \le j \le n$. This procedure appears ridiculous when key comparisons involve simple local variables; if a comparison requires a lengthy calculation or communication activity (see Topkis, 1986), the perspective changes.

By the Law of Large Numbers, this rule is asymptotically optimal for the above access model. We expect it should be also the best policy for any finite sequence of requests. If we charge both for comparisons and shuffles, there is little hope for an optimal policy with such a simple structure.

It is remarkable that *counter based* methods are not optimal with respect to our measure when the counters only reflect a limited portion of the past. This can be demonstrated on a model in which the relative order of the records after the mth request is determined by counters extracted from the reference history accumulated since the $l+1$st request, for some $1 \le l < m$.

Let $C^{(m-l)}$ be the partial frequency vector representing the last $m-l$ requests. Obviously, keeping the list in descending order of the counters in $C^{(m-l)}$ would not always minimize the expected access cost at the $m+1$st reference, as that would imply, for $l = m-1$, that

$$C_m(MTF) \le C_m(TR) \quad \forall m \ge 1 \; .$$

But the last inequality contradicts Rivest's proof (Rivest, 1976) that $C(MTF) > C(TR)$ for all non-trivial *rpv*'s.

ACKNOWLEDGMENT

The presentation of this paper has benefited much from the remarkably careful reading and detailed comments of an unknown referee. In particular, he suggested the use of permutations in a way that led to the current simple version of Lemmas 2 and 3 and to the present proof of the Theorem.

REFERENCES

Hofri M. and Shachnai H., Self-Organizing Lists and Independent References - a Statistical Synergy. The Department of Computer Science, the Technion TR#524, October 1988.

Lam, K., Siu, M.K., and Yu, C.T., A generalized counter scheme. *Theor. Comput. Sci.* **16,** #3, 271-278, 1981.

Rivest, R., On self-organizing sequential search heuristics. *Commun. ACM,* **19,** #2, 63-67, 1976.

Topkis, D.M., Reordering Heuristics for Routing in Communication Networks. *J. Appl. Prob. 23,* 1986, pp. 130-143.