LIST ORGANIZING STRATEGIES USING STOCHASTIC MOVE-TO-FRONT AND STOCHASTIC MOVE-TO-REAR OPERATIONS**

B. John Oommen*
E.R. Hansen+

SCS-TR-76

May 1985

*School of Computer Science, Carleton University, Ottawa, K1S 5B6, Canada

+Lockheed Missiles and Space Co. Inc., Sunnyvale, California 94086, U.S.A.

**Partially supported by the Natural Sciences and Engineering Research Council of Canada.

LIST ORGANIZING STRATEGIES USING STOCHASTIC MOVE-TO-FRONT AND STOCHASTIC MOVE-TO-REAR OPERATIONS+

B. John Oommen*

E.R. Hansen**

ABSTRACT

Consider a list of elements $\{R_1,\ldots,R_N\}$ in which the element R_i is accessed with an (unknown) probability s_i. If the cost of accessing R_i is proportional to i (as in sequently search) then it is advantageous if each access is accompanied by a simple reordering operation. This operation is chosen so that ultimately the list will be sorted in the descending order of the access probabilities.

In this paper we present two list organizing schemes -- the first of which uses bounded memory and the second which uses memory proportional to number of elements in the list. Both of the schemes reorder the list by moving only the accessed element. However, as opposed to the schemes discussed in the literature the move operation is performed stochastically in such a way that ultimately no more move operations are performed. When this occurs we say that the scheme has converged. We shall show that:

+ Partially supported by the Natural Sciences and Engineering Research Council of Canada.

* School of Computer Science, Carleton University, Ottawa, K1S 5B6, Canada.

** Lockheed Missiles and Space Co. Inc., Sunnyvale, California, 94086, U.S.A.

(i) The bounded memory stochastic move-to-front algorithm is expedient, but is always worse than the deterministic move-to-front algorithm.

(ii) The liner-memory stochastic move-to-rear scheme is optimal, independent of the distribution of the access probabilities. By this we mean that although the list could converge to one of its N! configurations, by suitably updating the probability of performing the move-to-rear operation, the probability of converging to the right arrangement can be made as close to unity as desired.

Key Words: Dynamic List Ordering, Move to Front Rule, Adaptive Learning, Self-Organizing Lists, Stochastic List Operations.

I. INTRODUCTION

Suppose we are given a set of elements $\{R_1,\ldots,R_N\}$. At every time instant one of these elements is accessed. Further, the element R_j is accessed with an unknown access probability s_j. We assume that the accesses are made independently. Whenever an element R_j is accessed, a sequential search is performed on the list. To minimize the cost of accessing, it is desirable that the records are ordered in the descending order of their access probabilities. We shall refer to a file ordered in this way as a completely organized file.

McCabe [11, pp.398-399; 12] was the first to propose a solution to this problem. His solution rendered the list dynamically self-organizing and it involved moving an element to the front of the list every time it was accessed. Using this rule, the Move-to-Front (MTF) rule, the limiting average value of the number of probes done per access has the value C_{MTF}, where,

$$C_{MTF} = \quad 0.5 + \sum_{i,j} (s_i \; s_j)/(s_i + s_j)$$

Many other researchers [3, 6-9,14] have also extensively studied the MTF rule and various properties of its limiting convergence characteristics are available in the literature.

McCabe [11, pp.398-399, 12] also introduced a scheme which is called the transposition rule. This rule requires that an accessed element is interchanged with its preceding element in the list, unless, of course, it is at the front of the list. Much literature is available on the transposition rule [1,2,4,11,12,13] but particularly important is the work of Rivest

[13] and Tenenbaum et al [1,15] who extensively studied this rule and suggested its generalizations - the Move-k-Ahead rule and the POS(k) rule. In the former, the accessed element is moved k-positions forward to the front of the list unless it is found in the first k positions - in which case it is moved to the front. The POS(k) rule moves the accessed element to position k of the list if it is in positions k+1 through n. It transposes it with the preceding element if it is in positions 2 through k. If it is the first element it is left unchanged.

Rivest [13] showed that the limiting behaviour of the transposition rule (quantified in terms of the average number of probes) was never worse than that of the MTF rule. He conjectured that the transposition rule has lower expected cost than any other reorganization scheme. He also conjectured that the move-k-ahead rule was superior to the move-k+1-ahead rule; but as yet this is unproven. Tenenbaum and Nemes [15] proved various results for the POS(k) rule primarily involving a distribution in which $s_2 = s_3 = \dots = s_N = (1-s_1)/(N-1)$. Their results seem to strengthen Rivest's conjecture.

All of the schemes discussed in the literature are represented by Markov chains which are ergodic. By virtue of this fact, the list can be in any one of its N! configurations - even in the limit. Thus, for example, a list which was completely organized can be thoroughly disorganized by the MTF rule by a <u>single</u> request for the element which is accessed most infrequently. Observe that after this unfortunate occurrence, it will take a long time for the list to be organized again - i.e., for this element to dribble its way to the tail of the list.

In this paper we propose two learning algorithms in which the elements of the list adaptively learn to find their place. The first algorithm is essentially a move-to-the-front algorithm with the exception that on the nth access, the accessed elements is moved to the front of the list with a probability f(n). This probability is systematically decreased every time an element is accessed. Ultimately on being accessed, each element tends to stay in the place where it is (as opposed to moving to the front of the list). In other words, the Markovian representation of the procedure is absorbing, as opposed to ergodic. The organization of the list gets "absorbed" into one of the N! orderings. But, we shall show that the scheme is expedient, i.e., if $s_i > s_j$, then the probability of absorption into an arrangement in which R_i precedes R_j is always greater than 0.5.

The second algorithm which we present is far more powerful. In this case, the algorithm is a move-to-rear scheme in which the accessed element R_j is moved to the <u>rear</u> of the list with a probability q_j. This quantity q_j is progressively decremented every time <u>the element</u> is accessed. As before, ultimately there is no move operation performed on the list. Thus this scheme too is absorbing in its Markovian representation and so could converge into any one of its N! arrangements. However, in this case, we shall show that the probability of converging to the optimal arrangement can be made as close to unity as desired.

The techniques introduced here require more workspace than the traditionally used methods. However, apart from the latter algorithm being much more accurate than all the algorithms reported in the literature, it is also computationally more efficient. This is because the access of an element requires the update of *exactly* one probability (namely the probability q_j associated with the accessed element R_j). Further, unlike the contemporary algorithms, since the list operations are essentially stochastic, a list operation is *not* necessarily performed on every access. Finally, since the Markovian representation of the scheme is absorbing, the number of list operations performed asymptotically decreases to zero.

As in the literature concerning the theory of adaptive learning, we shall use the terms algorithm, rule and scheme interchangeably. Further, for the sake of simplicity we shall assume that the access probabilities of the records are distinct. The cases when they are nondistinct are those when there are *many* optimal configurations. A remark about these cases will be made appropriately in the body of the paper.

II. BOUNDED MEMORY PROBABILISTIC MOVE-TO-FRONT OPERATIONS

The concept of performing probabilistic move operations on an accessed element is not entirely new. Kan and Ross [9] suggested a probabilistic transposition scheme and showed that no advantage was obtained by rendering the scheme probabilistic. Their scheme, however, required that the probability of performing the operation, be *time invariant*. As opposed to this, we shall define move operations which are essentially probabilistic, but the probabilities associated with the move operations are dynamically varied.

Let f(n) be the probability (at time 'n') of any element being moved to the front of the list on being accessed. Observe that this implies that an element, on being accessed, stays where it is with probability $(1-f(n))$. For an initial condition we define,

$$f(0) = a \qquad (2.1)$$

The probability $f(n)$ is updated every time any record is accessed. The updating scheme is given by (2.2) below, for $0<a<1$.

$$f(n+1) = a\, f(n) \quad \text{every time a record is accessed.} \qquad (2.2)$$

The quantity 'a' is defined as the updating constant.

Let ${}_iP_j(n)$ be the expected probability of record R_i succeeding R_j at the nth time instant. Clearly ${}_jP_i(n) = 1-{}_iP_j(n)$, for all n. We shall derive the transient and asymptotic properties of ${}_iP_j(n)$. To do this we need the following lemma.

LEMMA I

Let A be any nxn matrix with distinct eigenvalues. Let K be the matrix which diagonalizes A. Let

$$B(n) = I + a^n A$$

Then, B(n) is diagonalizable by the <u>same</u> matrix K, for all n.

Proof

Since A has distinct eigenvalues, and K diagonalies A,

$$K^{-1}\ A\ K = \mathrm{Diag}(\theta_1,\ldots,\theta_N)$$

where $\mathrm{Diag}(\theta_1,\ldots,\theta_N)$ is the diagonal matrix with the eigenvalues of A on its diagonal. Let $B(n) = I+a^n A$. Then,

$$\begin{aligned} K^{-1}\ B(n)\ K &= K^{-1}\ (I+a^n\ A)\ K \\ &= I + a^n\ K^{-1}\ A\ K \\ &= I + a^n \cdot \mathrm{Diag}(\theta_1,\ldots,\theta_N) \end{aligned}$$

and the lemma is proved.

Using the above lemma and the theory of Markov's chains we prove the following theorems.

THEOREM I

Let ${}_iP_j(n)$ be the expected probability of R_i succeeding R_j at the nth time instant. Then, ${}_iP_j(n)$ and ${}_jP_i(n)$ obey the following time varying Markov equation:

$$\begin{bmatrix} {}_iP_j(n+1) \\ \\ {}_jP_i(n+1) \end{bmatrix} = [B(n)] \begin{bmatrix} {}_iP_j(n) \\ \\ {}_jP_i(n) \end{bmatrix}$$

where $B(n) = \begin{bmatrix} 1-a^n s_i & a^n s_i \\ a^n s_j & 1-a^n s_j \end{bmatrix}^T$

Proof

R_i succeeds R_j at time instant 'n+1' if and only if

(a) R_i succeeded R_j at time instant 'n' and no list operation was performed, or,

(b) R_j was accessed and it was moved to the front of the list. Observe that R_i cannot succeed R_j if R_i was accessed and moved to the front.

Let ${}_ip_j(n)$ = Prob[R_i succeeds R_j at time 'n']. Clearly, $E[{}_ip_j(n)] = {}_iP_j(n)$. The above leads to the following recursive definition of ${}_ip_j(n)$.

$$
\begin{aligned}
{}_ip_j(n+1) &= 0 && \text{if } R_i \text{ accessed and MTF performed} \\
&= 1 && \text{if } R_j \text{ accessed and MTF performed} \\
&= {}_ip_j(n) && \text{otherwise.}
\end{aligned}
$$

Observe that the probabilities of the events defined above are readily available in terms of the unknown access probabilities. Further, a MTF operation is performed at 'n' with a probability a^n. Thus,

$$
\begin{aligned}
{}_ip_j(n+1) &= 0 && \text{w.prob. } s_i \cdot a^n \\
&= 1 && \text{w.prob } s_j\, a^n \\
&= {}_ip_j(n) && \text{w.prob } 1-(s_i+s_j)\, a^n
\end{aligned}
\tag{2.3}
$$

Taking conditional expectations, we have,

$$E[{}_ip_j(n+1) \mid {}_ip_j(n)] = a^n s_j + {}_ip_j(n) - {}_ip_j(n) \cdot a^n (s_i + s_j)$$

Taking expectations again and observing that $E[{}_ip_j(n)] = {}_iP_j(n)$, we get,

$${}_iP_j(n+1) = [1-a^n (s_i+s_j)]\, {}_iP_j(n) + s_j \cdot a^n$$

Since ${}_iP_j(n) + {}_jP_i(n) = 1$, we expand the constant term as

$$s_j\, a^n = s_j\, a^n\, [{}_iP_j(n) + {}_jP_i(n)]$$

Thus,

$${}_iP_j(n+1) = [1-a^n s_i]\; {}_iP_j(n) + [a^n s_j]\; {}_jP_i(n)$$

This leads to the following matrix equation

$$\begin{bmatrix} {}_iP_j(n+1) \\ \\ {}_jP_i(n+1) \end{bmatrix} = \begin{bmatrix} 1-a^n s_i & a^n s_j \\ \\ a^n s_i & 1-a^n s_j \end{bmatrix} \begin{bmatrix} {}_iP_j(n) \\ \\ {}_jP_i(n) \end{bmatrix}$$

and the theorem is proved.

THEOREM II

The constant matrix K, where

$$K = \begin{bmatrix} 1 & 1 \\ s_i/s_j & -1 \end{bmatrix}$$

diagonalizes B(n), for all values of n.

Proof

Observe that B(n) is of the form,

$$B(n) = I + a^n A$$

where

$$A = \begin{bmatrix} -s_i & s_j \\ s_i & -s_j \end{bmatrix}$$

Since B(n) is a stochastic matrix, we know that one of its eigenvalues is unity. Further, since the sum of the eigenvalues is equal to the trace of the matrix, the second eigenvalue is $1-a^n(s_i+s_j)$. Using Lemma I, we know that the matrix which diagonalizes B(0) also diagonalizes B(n) for all n. In this case, it is easy to see that the eigenvectors for B(0) are

(i) $[1 \quad s_i/s_j]^T$ for the eigenvalue unity, and

(ii) $[1 \quad -1]^T$ for the eigenvalue $1-(s_i+s_j)$

Thus, the constant matrix K, where,

$$K = \begin{bmatrix} 1 & 1 \\ s_i/s_j & -1 \end{bmatrix}$$

diagonalizes B(n) for all n. This proves the theorem.

Remark: By performing elementary operations, it is easy to see that K^{-1} has the form:

$$K^{-1} = \frac{1}{s_i+s_j} \begin{bmatrix} s_j & s_j \\ s_i & -s_j \end{bmatrix} \tag{2.4}$$

One can trivially verify that,

$$K^{-1} \; B(n) \; K = \mathrm{Diag}(1, 1-a^n(s_i+s_j)),$$

where,

$$\mathrm{Diag}(1,1-a^n(s_i+s_j) = \begin{bmatrix} 1 & 0 \\ 0 & 1-a^n(s_i+s_j) \end{bmatrix}$$

Similarly, K. $\mathrm{Diag}(1,1-a^n(s_i+s_j)) \cdot K^{-1}$ is exactly B(n).

THEOREM III

The value of ${}_iP_j(n)$ for an updating constant 'a' obtained by solving the Markov equation given by Theorem I, has the form:

$$ {}_iP_j(n) = \frac{s_j}{s_i+s_j} Q_{a,n} + \frac{s_i}{s_i+s_j} (1-Q_{a,n}) $$

where

$$ Q_{a,n} = 0.5 \quad [\prod_{k=0}^{n-1} (1-a(s_i+s_j))] $$

Proof

From the results of Theorem I, we can see that

$$ \begin{bmatrix} {}_iP_j(n+1) \\ \\ {}_jP_i(n+1) \end{bmatrix} = B(n) \begin{bmatrix} {}_iP_j(n) \\ \\ {}_jP_i(n) \end{bmatrix} $$

Thus, the solution of the matrix difference equation yields,

$$ \begin{bmatrix} {}_iP_j(n) \\ \\ {}_jP_i(n) \end{bmatrix} = B(n-1)\ B(n-2)...B(0) \begin{bmatrix} {}_iP_j(0) \\ \\ {}_jP_i(0) \end{bmatrix} = \prod_{k=0}^{n-1} B(k) \begin{bmatrix} {}_iP_j(0) \\ \\ {}_jP_i(0) \end{bmatrix} $$

Rewriting each B(k) in terms of the diagonal matrix Diag(1, $1-a^k(s_i+s_j)$),

$$ \begin{bmatrix} {}_iP_j(n) \\ \\ {}_jP_i(n) \end{bmatrix} = (\prod_{k=0}^{n-1} K.[\mathrm{Diag}(1,1-a^k(s_i+s_j))]\cdot K^{-1}) \cdot \begin{bmatrix} {}_iP_j(0) \\ \\ {}_jP_i(0) \end{bmatrix} $$

Since the product of each consecutive pair $K.K^{-1}$ yields the identity matrix, we write,

$$\begin{bmatrix} {}_iP_j(n) \\ \\ {}_jP_i(n) \end{bmatrix} = K[\prod_{k=0}^{n-1} \mathrm{Diag}(1, 1-a^k(s_i+s_j))] \cdot K^{-1} \cdot \begin{bmatrix} {}_iP_j(0) \\ \\ {}_jP_i(0) \end{bmatrix}$$

$$= K \cdot \begin{bmatrix} 1 & 0 \\ \\ 0 & \prod_{k=0}^{n-1} (1-a^k(s_i+s_j)) \end{bmatrix} K^{-1} \begin{bmatrix} {}_iP_j(0) \\ \\ {}_jP_i(0) \end{bmatrix} \tag{2.5}$$

Let $Q_{a,n} = 1/2 \prod_{k=0}^{n-1} (1-a^k(s_i+s_j))$. Since, with no loss of generality, ${}_iP_j(0) = {}_jP_i(0) = 0.5$, we expand (5) above to yield,

$$\begin{bmatrix} {}_iP_j(n) \\ \\ {}_jP_i(n) \end{bmatrix} = \frac{1}{s_i+s_j} \begin{bmatrix} 1 & 1 \\ \\ \frac{s_i}{s_j} & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \\ 0 & 2 \cdot Q_{a,n} \end{bmatrix} \begin{bmatrix} s_j & s_j \\ \\ s_i & -s_j \end{bmatrix} \begin{bmatrix} 0.5 \\ \\ 0.5 \end{bmatrix}$$

After considerable simplification this results in

$$\begin{bmatrix} {}_iP_j(n) \\ \\ {}_jP_i(n) \end{bmatrix} = \frac{1}{s_i+s_j} \begin{bmatrix} s_j\ (1-Q_{a,n}) + s_i\ Q_{a,n} \\ \\ s_i\ (1-Q_{a,n}) + s_j\ Q_{a,n} \end{bmatrix} \tag{2.6}$$

and the theorem is proved.

Remark: Observe that ${}_iP_j(n) = {}_jP_i(n) = 0.5$ if $s_i = s_j$. This is intuitively satisfying.

To prove the asymptotic value of ${}_iP_j(n)$ we need the following lemma.

Lemma II

The infinite product:

$$\prod_{k=1}^{\infty} (1+b_k) \qquad (\text{where } b_k \neq -1 \text{ for all } k)$$

tends to a <u>non-zero</u> <u>finite</u> limit if and only if the infinite sum,

$$\sum_{k=1}^{\infty} b_k$$

is convergent.

Proof

The lemma is proved in Titchmarsh [17, pp.13-15].

THEOREM IV

The stochastic bounded memory Move-to-Front Algorithm is asymptotically always less accurate than the deterministic Move-to-Front Algorithm.

Proof

Consider the term for ${}_iP_j(n)$ as

$${}_iP_j(n) = \frac{s_i}{s_i+s_j}(1-Q_{a,n}) + \frac{s_j}{s_i+s_j} Q_{a,n}$$

where $Q_{a,n}$ is defined in Theorem III.

Using Lemma II, it is clear that $Q_{a,n}>0$ as $0<a<1$, but tends to zero as a tends to unity. Differentiating with respect to a, we obtain,

$$\frac{\delta\, {}_iP_j(n)}{\delta a} = (s_j - s_i) \cdot Q_{a,n} \cdot \sum_{k=1}^{n-1} \frac{k \cdot a^{k-1}}{1-a^k\,(s_i+s_j)}$$

Since $0<a<1$, and $0<s_i+s_j<1$, we have,

$$0 < 1-a^k\,(s_i+s_j) < 1 \qquad \text{for all } k \geq 1.$$

Assume that with no loss of generality that $s_i < s_j$. This tells us that,

$$\frac{\delta\, {}_iP_j(n)}{\delta a} > 0 \tag{2.7}$$

In other words, ${}_iP_j(n)$ has no stationary point with respect to a in the interval $0 < a < 1$. Further, due to (2.7), the largest value of ${}_iP_j(n)$ occurs when a=1. Since (2.7) is true for finite and infinite values of n, the value of ${}_iP_j(\infty)$ is maximized at the largest acceptable value of a and the theorem is proved.

Corollary IV.1

The stochastic bounded memory Move-to-Front Algorithm is expedient independent of the access distribution of the records.

Proof

Due to the multiplying factor of 0.5, and the previous theorem, it is easy to see that for all a, $0<Q_{a,n}<0.5$. The result is now obvious since ${}_iP_j(n)$ is merely a convex combination of $s_i/(s_i+s_j)$ and $s_j/(s_i+s_j)$ weighted by $(1-Q_{a,n})$ and $Q_{a,n}$ respectively.

Remark: Throughout this discussion it was assumed that the single memory location that stores f(n) can contain an arbitrarily small positive real number. In practice, however, all that we need to store is an index, n, of the time that has lapsed since the file reorganization scheme was initiated. From this index, f(n) can be computed trivially, since,

$$f(n) = a^n$$

It is also appropriate to observe that the maximum number that this index should attain is governed by the uniform random number generator accessible to the system. If the smallest positive number yielded by the random number generator is x_{min}, then the memory location which stores n need not store numbers larger than n_{max}, where,

$$n_{max} = \lceil \log_a (x_{min}) \rceil$$

We now proceed to study the linear memory stochastic move-to-rear scheme.

III. LINEAR MEMORY PROBABILISTIC MOVE-TO-REAR OPERATIONS

In this section we shall show how a probabilistic scheme with an absorbing Markovian representation can indeed be made asymptotically optimal. The idea is essentially one of moving the accessed element R_j to the rear of the list with a probability q_j which is systematically updated so that, as in the previous case, in the limit, no list operations are performed. In this case, we shall show that the list converges to one of the N! possible arrangements, and the probability of converging to the optimal one can be made arbitrarily close to unity. Let $P_j(n)$ be the probability (at time 'n') of moving the element R_j to the rear of the list on being accessed. This implies that on being accessed, the element stays where it is with probability

$(1-q_j(n))$.

Initially, we set $q_j(0)=1$ for all $j=1,\ldots,N$. After this, the quantity $q_j(n)$ is decremented every time R_j is accessed as below:

$$
\begin{aligned}
q_j(n+1) &= aq_j(n) \text{ if } R_j \text{ is accessed.} \\
&= q_j(n) \quad \text{otherwise.}
\end{aligned}
\tag{3.1}
$$

Observe that the updating scheme is reiminiscent of the Linear Reward-Inaction scheme studied extensively in the area of adaptive learning.

We now derive the properties of $q_j(n)$. To simplify notation, unless explicitly stated, q_j will refer to the quantity $q_j(n)$.

THEOREM V

$E[q_i(n)]$ decreases monotonically with time. Further, for all $j \neq k$,

$$s_j > s_k \text{ if and only if } E[q_j(n)] < E[q_k(n)]. \tag{3.2}$$

Proof

Consider the random variable $q_i(n+1)$. By virtue of (3.2), the latter has the following distribution:

$$
\begin{aligned}
q_i(n+1) &= aq_i, \qquad \text{w.prob. } s_i \\
&= q_i \qquad \text{w.prob } (1-s_i)
\end{aligned}
\tag{3.3}
$$

Thus,
$$
\begin{aligned}
E[q_i(n+1) \mid q_i] &= aq_i s_i + q_i(1-s_i) \\
&= q_i(1-(1-a)s_i)
\end{aligned}
$$

$$= e_i q_i, \qquad \text{where } e_i = 1-(1-a)s_i \qquad (3.4)$$

Note that since $0<a<1$, for all i, e_i obeys $0<e_i<1$. Taking expectations again, we obtain,

$$E[q_i(n+1)] = e_i E[q_i(n)]. \qquad (3.5)$$

The difference equation (5) subject to the initial condition $q_i(0)=1$, yields the solution:

$$E[q_i(n)] = e_i^n.$$

Clearly, $E[q_i(n)]$ is monotonically decreasing with n. Further,

$$s_j > s_k \iff e_j < e_k \iff e_j^n < e_k^n,$$

and the theorem is proved.

Corollary V.1

For all i,

$$\lim_{n\to\infty} q_i(n) = 0 \qquad \text{w.prob 1.}$$

The result follows since the Markov process $\{q_i(n)\}$ has only one absorbing barrier, namely the probability 0 [10].

THEOREM VI

For n>0, $Var[q_i(n)]$ decreases monotonically with n and its limiting value is zero.

Proof

From the distribution of $q_i(n)$ given by (3.3), we obtain, converge w.p.1 to zero. Further, at any instant,

$$E[q_j(n)] > E[q_k(n)] \text{ if and only if } s_j < s_k.$$

Observe that this is true for all $0<a<1$. By appropriately choosing the value of the updating constant 'a', we shall now show that a stochastically <u>stronger</u> inequality exists - which not merely relates the <u>expected values</u> of q_j and q_k but the probabilities themselves.

THEOREM VII

For all j, k where $j \neq k$, if $s_j > s_k$, then, the quantity $Pr[q_j(n) < q_k(n)]$ can be made as close to unity as desired.

Proof:

Assume with no loss of generality that $s_j > s_k$. Let,

$$x_{j,k}(n) = q_j(n) / q_k(n) .$$

Clearly, $x_{j,k}(0)=1$. Further, $x_{j,k}(n)$ can only assume non-negative values. Consider the distribution of $x_{j,k}(n+1)$ given the values of $q_j(n)$ and $q_k(n)$. By virtue of (3.1),

$$\begin{aligned} x_{j,k}(n+1) &= (a\, q_j) / q_k && \text{w. prob. } s_j, \\ &= q_j / (a\, q_k) && \text{w. prob. } s_k, \\ &= q_j / q_k && \text{w. prob. } (1-s_j-s_k). \end{aligned}$$

Thus,
$$\begin{aligned} x_{j,k}(n+1) &= a\, x_{j,k}(n) && \text{w. prob. } s_j, \\ &= x_{j,k}(n) / a && \text{w. prob. } s_k, \\ &= x_{j,k}(n) && \text{w. prob. } (1-s_j-s_k). \end{aligned} \quad (3.8)$$

Taking conditional expectations yields,

$$E[x_{j,k}(n+1) \mid x_{j,k}(n)] = 1/a\ [a^2 s_j + a(1-s_j-s_k)+s_k]\ x_{j,k}(n).$$

Whence on taking expectation again and observing that $x_{j,k}(0)=1$, we get,

$$E[x_{j,k}(n)] = (h_{j,k})^n, \text{ where } h_{j,k} = 1/a\ [a^2 s_j + a(1-s_j-s_k)+s_k]. \tag{3.9}$$

Let a be any real number satisfying, $s_k < a\ s_j$. Then,

$$s_k/a < s_j$$
$$\Rightarrow s_k(1-a)/a < s_j(1-a)$$
$$\Rightarrow s_k(1/a - 1) < s_j(1-a)$$
$$\Rightarrow as_j - s_j - s_k + (s_k/a) < 0$$
$$\Rightarrow h_{j,k} < 1.$$

Thus $h_{j,k}$ can be made strictly less than unity by appropriately choosing 'a'. From (3.9), this implies that $E[x_{j,k}(n)]$ can be rendered monotonically <u>decreasing</u> with n, and further,

$$\lim_{n\to\infty} E[x_{j,k}(n)] = 0$$

But $x_{j,k}(n)$ is always nonnegative. This implies that

$$\lim_{n\to\infty} x_{j,k}(n) \to 0 \qquad \text{w. prob. 1}$$

Thus, $\lim \Pr[x_{j,k}(n) > 0] \to 0$ w. prob. 1,

and the result follows.

Remark: Observe that although the values of $\{s_i\}$ are unknown, the above theorem says that by making 'a' sufficiently close to unity the asymptotic value $\Pr[q_j(n) < q_k(n)]$ can be made as close to zero as desired.

We now prove the optimality of the scheme.

Let ${}_iY_j(n)$ be the expected probability that R_i succeeds R_j at the nth time instants. Clearly,

$${}_jY_i(n) = 1 - {}_iY_j(n) \text{ for all } i,j=1,\ldots N \;;\; i \neq j.$$

The properties of ${}_iY_j(n)$ are summarized by the following theorem.

THEOREM VIII

The asymptotic value of ${}_iY_j(n)$ is unity if and only if $s_i > s_j$.

Proof

The theorem follows in a manner analogous to that of Theorem I.

Note that R_i succeeds R_j at time 'n+1' if and only if:

(a) R_i succeeded R_j at time n and no list operation was performed, or

(b) R_i was accessed at time n and it was moved to the rear of the list.

Let ${}_iy_j$ = Prob[R_i succeeds R_j at time 'n']. Clearly, $E[{}_iy_j(n)] = {}_iY_j(n)$. We thus have the following recursive definition of ${}_iy_j(n)$.

$$\begin{aligned} {}_iy_j(n+1) &= 0 \text{ if } R_j \text{ was accessed and MTR is performed.} \\ &= 1 \text{ if } R_i \text{ was accessed and MTR is performed.} \\ &= {}_iy_j(n) \text{ otherwise.} \end{aligned}$$

Let Z_i be the number of times R_i was accessed prior to and including the time instant 'n'. Clearly, Z_i is a random variable, with $\sum_{i=1}^{N} Z_i = n$. Further, the probability of moving

R_i at the nth time instant (on being accessed) to the rear of the list is exactly a^{Z_i-1}. Thus, we rewrite ${}_iy_j(n)$ to obey the following equations:

$$
\begin{aligned}
{}_iy_j(n+1) &= 0 && \text{w.prob. } s_j \cdot a^{Z_j-1} \\
&= 1 && \text{w.prob. } s_i\, a^{Z_i-1} \\
&= {}_iy_j && \text{otherwise.}
\end{aligned}
$$

Taking expectations twice and observing that $E[{}_iy_j(n)]={}_iY_j$, we obtain,

$$
\begin{bmatrix} {}_iY_j(n+1) \\ {}_jY_i(n+1) \end{bmatrix} = \begin{bmatrix} 1-a^{Z_j-1} s_j & a^{Z_j-1} s_j \\ a^{Z_i-1} s_i & 1-a^{Z_i-1} s_i \end{bmatrix}^T \begin{bmatrix} {}_iY_j(n) \\ {}_jY_i(n) \end{bmatrix}
$$

Unlike the case of Theorem II, this time varying difference equation cannot be solved directly using the theory of diagonalization. This is because there is no single constant matrix which diagonalizes the Markov matrix for every time instant. However, we consider the asymptotic value of ${}_iY_j(n)$. On converging,

$$
{}_iY_j(\infty) = a^{Z_i-1} s_i + (1- a^{Z_i-1} s_i - a^{Z_j-1} s_j)\ {}_iY_j(\infty)
$$

which yields:

$$
{}_iY_j(\infty) = \lim_{Z_i,Z_j\to\infty} \left[a^{Z_i-1}\cdot s_i/(a^{Z_i-1}\cdot s_i + a^{Z_j-1}\cdot s_j) \right] = \lim_{Z_i,Z_j\to\infty} \left[\frac{a^{Z_i} s_i}{a^{Z_i} s_i + a^{Z_j} s_j} \right]
$$

Let $t = s_i/s_j$. Further, using the law of large numbers,

$$Z_j = s_j/s_i \,.\, Z_i = t\ Z_i$$

Thus, $${}_iY_j(\infty) = \lim_{Z_i \to \infty} \left[\frac{1}{1 + t\, a^{(t-1)Z_i}} \right]$$

Now, since we assume that $s_i < s_j$, t is strictly less than unity.

Therefore,

$$\lim_{Z_i \to \infty} a^{(t-1)Z_i} = 0$$

and hence, $\lim_{Z_i \to \infty} {}_iY_j(\infty) = 1.$

Hence the theorem!

Remark: Observe that in this case too, ${}_iY_j(\infty)$ has a value of 0.5 if $s_i = s_j$, which is what we would expect.

A natural consequence of the above theorem is the corollary stated below. The corollary is obvious in as much as the above theorems are valid for every arbitrary pair of records R_i and R_j.

COROLLARY VIII.1

The probability of the list converging to the optimal arrangement (out of the N! possible arrangements) can be made as close to unity as desired.

Remarks: (1) Just as in the case of the stochastic MTF scheme, observe that we have assumed that the quantities $q_j(n)$ can be arbitrarily small positive real numbers. However, in practice, just as in the above case, it is sufficient to retain indices which remember the number of times a record is accessed. From these indices the quantities $q_j(n)$ can be trivially computed. Observe too that the maximum integer a memory location should contain need not exceed n_{max}, where,

$$n_{max} = \lceil \log_a x_{min} \rceil + 1$$

where x_{min} is the smallest positive real number yielded by the random number generator accessible to the user.

(2) In this connection, it is beneficial to compare the scheme we have proposed with the scheme which keeps the records <u>sorted</u> in the order of the number of times they have been accessed. First of all, observe that our scheme does not require any "global" comparisons, as for example, comparing $q_i(n)$ with $q_j(n)$. Further consider the case when two records R_i and R_j have exactly the same access probabilities. If Z_i and Z_j respectively are the number of times these records are accessed, then, notice that if one keeps the list sorted on the basis of counters,

$$\Pr[\text{Move Operation Performed}] \mid Z_i = Z_j;\ R_j \text{ succeeds } R_i;\ R_j \text{ accessed}] = 1.$$

Notice too that this probability will be unity <u>even in the limit</u>. However, the latter probability will tend to <u>zero</u> in the stochastic MTR scheme which we have proposed essentially because of the absorbing Markovian representation of our scheme. One can easily extrapolate this situation to the case when many records have been accessed exactly the same number of times.

CONCLUSIONS

We have considered a list of elements $\{R_1, \ldots, R_N\}$ in which the element R_i is accessed with a probability s_i, which is unknown apriori. Two stochastic list organizing schemes have been proposed both of which have absorbing Markovian representations. The first of these schemes requires bounded memory and is of a move-to-front flavour. It is expedient, but is always less optimal than the deterministic move-to-front algorithm. The second scheme performs a move to the rear operation on the accessed element R_j with a probability q_j which is systematically decreased. Ultimately, the list gets absorbed into one of the N! possible arrangements. We have shown that the asymptotic probability of converging to the optimal arrangement can be made as close to unity as desired.

ACKNOWLEDGEMENTS

I am greatly indebted to my colleague Nicola Santoro who introduced me to the problem and to the literature in the field. I am grateful to him and my colleague Mike Atkinson for their encouragement and comments during the course of the study. I am also grateful to Professor Ian Munroe, from the University of Waterloo who first conjectured, while in a personal conversation, the result that the Stochastic MTF scheme was at its best expedient.

[16] Oommen, B.J., "On the Use of Smoothsort and Stochastic Move-to-Front Operations for Optimal List Organization", Proc. of the Twenty-Second Allerton Conference on Communication, Control and Computing, October 1984, pp.243-252.

[17] Titchmarsh, E.C., The Theory of Functions, Oxford University Press, 1964.

REFERENCES

[1] Arnow, D.M. and Tenebaum, A.M., "An Investigation of the Move-Ahead-k Rules", Congressus Numerantium, Proc. of the Thirteenth Southeastern Conference on Combinatorics, Graph Theory and Computing, Florida, February 1982, pp.47-65.

[2] Bitner, J.R., "Heurishics That Dynamically Organize Data Structures", SIAM J. Comput., Vol.8, 1979, pp.82-110.

[3] Burville, P.J. and Kingman, J.F.C., "On a Model for Storage and Search", J. Appl. Probability, Vol.10, 1973, pp.697-701.

[4] Cook, C.R., and Kim, D.J., "Best Sorting Algorithm for Nearly Sorted Lists", Comm. ACM, Vol.23, 1980, pp.620-624.

[5] Dijkstra, E.W., "Smoothsort, An Algorithm for Sorting in SITU", Science of Computer Programming, 1982, pp.223-233.

[6] Gonnet, G.H., Munro, J.I. and Suwanda, H., "Exegesis of Self Organizing Linear Search", SIAM J. Comput., Vol. 10, 1981, pp.613-637.

[7] Hendricks, W.J., "The Stationary Disribution of an Interesting Markov Chain", J. App. Probability, Vol.9, 1972, pp.231-233.

[8] Hendricks, W.J., "An Extension of a Theorem Concerning an Interesting Markov Chain", J. App. Probability, Vol.10, 1973, pp.231-233.

[9] Kan, Y.C. and Ross, S.M., "Optimal List Order Under Partial Memory Constraints", J. App. Probability, Vol.17, 1980, 1004-1015.

[10] Karlin, S. and Taylor, H.M., "A First Course in Stochastic Processes", Academic Press, 1975.

[11] Knuth, D.E., "The Art of Computer Programming, Vol.3, Sorting and Searching", Addison-Wesley, Reading, Ma, 1973.

[12] McCabe, J., "On Serial Files With Relocatable Records", Operations Research, Vol.12, 1965, pp.609-618.

[13] Rivest, R.L., "On Self-Organizing Sequential Search Heuristics", Comm. ACM, Vol.19, 1976, pp.63-67.

[14] Sleator, D. and Tarjan, R., "Amortized Efficiency of List Update Rules", Proc. of the Sixteenth Annual ACM Symposium on Theory of Computing, April 1984, pp.488-492.

[15] Tenenbaum, A.M. and Nemes, R.M., "Two Spectra of Self-Organizing Sequential Search Algorithms", SIAM J. Comput., Vol.11, 1982, pp.557-566.