University of Waterloo
Department of Computer Science
Waterloo, Ontario N2L 3G1

November 9, 1988

Ambard/Innolenti
Unite D'Enseignement
Et De Recherche Scientifique De Luminy
Mathematique - Informatique
Case 901
70, Route Leon-Lachamp
13288 Marseille Cedex 9
FRANCE

INVOICE

REPORT(S) ORDERED

CS-80-07

TOTAL COST $4.00 My account number is 126-4114-02 and the bank is the Canadian Imperial Bank of Commerce, Campus Centre Branch, University of Waterloo, Waterloo, Ontario. Canada N2L 3G1

Would you please make your cheque or international bank draft payable to the Computer Science Department, University of Waterloo and forward to my attention.

Thanking you in advance.

Yours truly,

Susan DeAngelis

Susan DeAngelis (Mrs.)
Research Report Secretary
Computer Science Dept.

Received May 19/89 Banque Nationale de Paris Toronto, Ont.

/sd

[redacted]

Gérald AMBARD, Librairies

[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]

France

Marseilles, le 12/9/1989

to
Mrs. Susan De Angelis

Dear Mrs De Angelis,

Could you please indicate a bank name and your account in this bank in order to pay you promptly.

Thank you in Advance

Sincerely Yours

1. please bring up to date your new invoice
2. Xeroscopy of your invoice enclosed

8345?

University of Waterloo
Department of Computer Science
Waterloo, Ontario N2L 3G1

July 5, 1988

INVOICE

Ambard/Innolenti
Unite D'Enseignement
Et De Recherche Scientifique De Luminy
Mathematique - Informatique
Case 901
70, Route Leon-Lachamp
13288 Marseille Cedex 9
FRANCE

REPORT(S) ORDERED

CS-80-07

TOTAL COST $4.00

Please send an international bank draft not a money order. Thank you.

Would you please make your cheque or international bank draft payable to the **Computer Science Department, University of Waterloo** and forward to my attention.

Thanking you in advance.

Yours truly,

Susan DeAngelis

Susan DeAngelis (Mrs.)
Research Report Secretary
Computer Science Dept.

/sd

EXEGESIS OF SELF-ORGANIZING LINEAR SEARCH*

GASTON H. GONNET†, J. IAN MUNRO† AND HENDRA SUWANDA‡

Abstract. We consider techniques for self-organizing linear search, examining the behavior of methods under arbitrary and specific probability distributions.

The notion of moving an element forward after it has been accessed k times in a row is introduced. One implementation performs the transformation after any k identical requests. A second essentially groups requests into batches of k, and performs the action only if all requests of a batch are the same. Adopting as the transformation, the move to front heuristic, the second approach is shown in general to be superior. We show that the batched approach, with $k = 2$, leads to an average search time no greater than 1.21... times that of the optimal ordering. For the more direct approach, a ratio of 1.36... is shown under the same constraints.

The simple move to front heuristic (i.e., $k = 1$) is also examined. It is shown that for a particular distribution this scheme can lead to an average number of probes $\pi/2$ times that of the optimal order. Within an interesting class of distributions, this is shown to be the worst average behavior.

Key words. self organizing files, linear search, move to front, transpose rule, complexity analysis, asymptotic analysis, heuristics

1. Introduction and preliminary results. Suppose we have a file which must be searched sequentially, and that the probabilities of accessing the various elements are fixed and independent, but unfortunately, unknown. The obvious approach to the problem of finding a good ordering for the list is to count requests and dynamically keep the file in decreasing order by request count. Given enough time, by the law of large numbers we will clearly arrive at the best possible ordering. The cost of maintaining such counters is very often prohibitive, and so heuristics for rearranging the file without the use of extra ordering information have been studied (Bitner [3], Hendricks [8], Knuth [9], McCabe [10], Rivest [11], Tanenbaum [12]). The basic approach of such methods is to consider a set of n permutations (for a list of length n), $\pi_1, \ldots, \pi_n$. If the element currently in position i is requested, then π_i is applied to the list. We will call such a technique a *memory-free* self-organizing heuristic. The most obvious such heuristic is the *transposition* of the requested element with the one in front of it. Another, more drastic approach is to *move* the requested element *to* the *front* of the list, and effectively slide the others back one position. This method, while asymptotically not as effective as the transposition rule, has proven more amenable to analysis.

Our contribution is, first, to continue the study of such memory-free heuristics, comparing their behavior with that of an optimally ordered list. Secondly, but perhaps more significantly, we show that the use of even a very small amount of storage, to "remember" the location of records which have been requested, can lead to expected search costs arbitrarily close to that of the optimal ordering. The pre-

Received by the editors January 3, 1980, and in revised form February 2, 1981.

*This work was supported by the Natural Sciences and Engineering Research Council of Canada under grants A8237 and A3353. This paper was typeset at the University of Waterloo.

† Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1.

‡ Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, M5S 1A7.

denotes (the asymptotic value of) expected number of probes under the transposition rule, $F(\vec{p})$ the expected number under the move to front rule, and $Opt(\vec{p}) = \sum ip_i$ the expected number under the optimal ordering, then

$$F(\vec{p}) = 2\sum_{i=1}^{n}\sum_{j=1}^{i}\frac{p_i p_j}{p_i+p_j}$$

and for all nontrivial distributions

$$T(\vec{p}) < F(\vec{p}) < 2Opt(\vec{p})$$

(iii) While the move to front heuristic is known to produce a system with cost no more than twice that of the optimal ordering, the greatest value of $F(\vec{p})/Opt(\vec{p})$ which has been demonstrated is $2\ln 2(\approx 1.386)$. This occurs under Zipf's law, i.e. $p_i = 1/(iH_n)$ (where H_n denotes the n^{th} Harmonic number. (Knuth [6]).

(iv) If any single memory free heuristic has asymptotic behaviour at least as good as every other such method for every probability distribution, it is the transposition rule (A. Yao as reported by Rivest [8]).

A casual glance at the results cited above indicates that the measure of effectiveness of a heuristic which has been used primarily is the (asymptotic) ratio of the expected behaviour of the heuristic to that of the optimal ordering. This seems appropriate in studying classes of distributions under which the average cost of searching an optimally ordered list is not bounded by a constant as the list becomes longer (e.g. Zipf's law). It is not clear that this is a good measure for specific distributions, such as $p_i = \frac{1}{2}^i$, under which the expected search time is bounded by a constant. Indeed we feel, and in section 3 give "evidence", that this latter case may be the reason that we do not as yet have tight bounds on the behaviour of even the move to front heuristic relative to the optimal ordering.

Yao's observation (that if any memory-free heuristic is better than all others over all possible distributions, it is the transposition rule) is perhaps the most intriguing fact known about the self organization of linear files. A definition of optimality for such a class must, however, be made carefully. Rivest [8] informally suggested such a definition, saying that a set of permutations was an optimal heuristic if, over all distributions and all initial orderings of the file, the expected number of probes to perform a search under the heuristic was (asymptotically) no greater than that under any other scheme using the same distribution and initial configuration. This wording is, unfortunately, a bit too strong, in that the "do nothing" heuristic outperforms all others (under most distributions) if it is fortunate enough to find its keys in decreasing order of probability. There is, then, under this definition, no optimal memory-free heuristic. We prefer to make a slight modification saying: A set of permutations is an *optimal memory free heuristic* if for all probability distributions the maximum over all initial configurations of the asymptotic value of the average search cost is no greater than that for any other such scheme.

A slightly weaker, but equally satisfactory, definition is obtained by insisting that the expected asymptotic behaviour be independent of the initial configuration, which appears to have been Rivest's intention.

Under either definition the theorems of Rivest and observations of Yao regarding an optimal heuristic hold.

Next to the asymptotic behaviour of a heuristic, its most interesting property is the rate at which it converges to this asymptotic behaviour. Based on the work of Bitner [3], it can be shown that while the transposition rule is asymptotically better than the move to front rule, the former can require $\Omega(n^2)$ probes to converge to within a factor of $(1+\epsilon)$ of the asymptotic behaviour from an arbitrary initial configuration. With the following definition, we can make a more general statement about such methods.

Let G denote the class of all memory-free self-organizing heuristics which

(i) When the element in position i is requested, that element is moved to position τ_i ($\tau_i < i$ if $i \neq 1$ and $\tau_1 = 1$) all elements in positions τ_i through $i-1$ are moved back one position

(ii) No other elements are moved.

(iii) If $i < j$ then $\tau_i \leqslant \tau_j$.

THEOREM 1.1. *The asymptotic behaviour of a reorganization technique in G is independent of the initial configuration.*

Proof. First we ignore the elements with probability 0 of being accessed since they will eventually percolate to the end of the list and be of no concern. We note that any configuration is reachable after at most n^2 accesses (we simply access each element enough times to move it to position 1 in the reverse of the desired order). The probability of such a sequence of requests is non zero, consequently given enough time, each configuration is reachable with probability 1. The theorem then follows. □

THEOREM 1.2. *If H_1 and H_2 are heuristics in G such that $H_1 \neq H_2$ and $\tau_i(H_1) \leqslant \tau_i(H_2)$ for all i, then, for all nontrivial distributions, H_1 converges to its asymptotic behaviour more quickly than H_2, but the expected search time under H_2 is, asymptotically, less than that of H_1.*

Proof. The proof is similar to the one in [8].

THEOREM 1.3. *The transposition rule can take $\Omega(n^2)$ accesses to reach within factor $1+\epsilon$ of the steady state behaviour.*

Proof. To construct such an example, let n be the total number of elements, k of which have accessing probabilities $\frac{1-\delta}{k}$ and the $n-k$ remaining have probability $\frac{\delta}{n-k}$.

Let δ be small enough so that $\frac{\delta}{n-k} < \frac{1-\delta}{k}$, then the cost of the optimal configuration is

$$Opt(\vec{p}) = \frac{k(k+1)(1-\delta)}{2k} + \frac{n(n+1)-k(k+1)}{2} \cdot \frac{\delta}{n-k}.$$

The worst case (the reverse of the optimal) has a cost

$$\text{worst case} = \frac{n(n+1)-(n-k)(n-k+1)}{2} \cdot \frac{1-\delta}{k} + \frac{(n-k)(n-k+1)}{2} \cdot \frac{\delta}{n-k}.$$

Each access to the file either maintains the average cost, or increases or decreases it by $\frac{1-\delta}{k}-\frac{\delta}{n-k}$.

Since the steady state cost of the transposition rule is less than twice the optimal, the least number of accesses to reach a factor of $1+\epsilon$, $(0<\epsilon<1)$ of the steady state is

$$\min accesses \geqslant \frac{worst\ case-2(1+\epsilon)Opt(\vec{p})}{\frac{1-\delta}{k}-\frac{\delta}{n-k}}$$

$$\geqslant (n-\frac{1}{2}-\epsilon)k-(\frac{3}{2}+\epsilon)k^2+O(\delta)$$

iff $k = \alpha n$ then for any α such that $\alpha-\frac{5}{2}\alpha^2>0$ or $0<\alpha<\frac{2}{5}$ then

$$\min accesses = \Omega(n^2) \quad \square$$

For the move to front rule we define the overwork as in Bitner [3] to be the excess work, from the steady state situation, to be done at time t.

We know that

$$Ow(t) = \frac{1}{2}\sum_{1\leqslant i<j\leqslant n}\frac{(p_i-p_j)^2}{p_i+p_j}(1-p_i-p_j)^t.$$

THEOREM 1.4. *The overwork at time t in the move to front rule is $O(n^2/t)$.*

Proof. Using the fact that for the chosen summation range $p_j \leqslant p_i$, we can rewrite the above as

$$Ow(t) \leqslant \sum_{i=1}^{n} p_i(1-p_i)^t(n-i).$$

For a given t, $p_i(1-p_i)^t \leqslant \frac{1}{t+1}\left(\frac{t}{t+1}\right)^t \leqslant \frac{e^{-1}}{t+1}$ and consequently

$$Ow(t) \leqslant \frac{e^{-1}(n-1)n}{4(t+1)} = O(\frac{n^2}{t}). \quad \square$$

This bound is tight in the sense that we can find a file such that for any $\epsilon>0$, $Ow(t) = \Omega(n^{1-\epsilon})$ for $t = o(n^{1+\epsilon})$. Consider the table with

$$p_i = \frac{2n-i}{n^{2+\epsilon}} \qquad 1<i\leqslant n/2$$

$$p_i = \frac{2n-i}{n^{3+\epsilon}} \qquad n/2<i\leqslant n$$

and

$$p_1 = 1-\sum_{i>1} p_i$$

with n large enough such that $p_1>p_2$. Then

$$Ow(t) \geqslant \frac{1}{2}\sum_{i=2}^{n/2}\sum_{j>n/2}\frac{(p_i-p_j)^2}{p_i+p_j}(1-p_i-p_j)^t$$

$$\geqslant \frac{(n-1)(n-2)}{8}\Theta\left(\frac{1}{n^{1+\epsilon}}\right)\left(1-\Theta\left(\frac{1}{n^{1+\epsilon}}\right)\right)^t$$

For $t = o(n^{1+\epsilon})$ we derive

$$Ow(t) = \Omega(n^{1-\epsilon}).$$

For this example we also find that

$$\frac{Ow(t)}{F(\vec{p})} = \Omega(1).$$

2. k in a Row Heuristics. As we have noted, keeping a count on the number of accesses made on each element does enable us to order a table optimally. The objection, of course, is the storage requirement. In this section we proposed a class of heuristics, closely akin to the memory-free techniques, which use ($\log n + \log k$), (for fixed k), extra bits of storage rather than the $\Theta(n)$ or more bits required for counter schemes. These techniques yield near optimal behaviour.

The basic approach is simple, we apply the transposition (move to front or any other) heuristic only if the same element is accessed k times in a row. We first analyze the simple k heuristic and later a slight modification of it.

2.1. Simple k Heuristics. We first analyse the simple k heuristic with the move to front rule. Let $h_k(j,i)$ denote the (asymptotic) probability that record j precedes record i in the list. This value can be found by considering the Markov chain shown in figure 1.

FIGURE 1
Markov Chain for the simple k heuristic.

i_0 is the initial state where record i precedes record j. In this state, record j is accessed with probability p_j and causes a transition to state i_1. Otherwise, with probability $1-p_j$ we remain in state i_0. If record j is accessed k times in a row, the move to front rule is applied and we will arrive at state j_0 where record j precedes record i. The probability of record j preceding record i in the list is given by the sum of the probabilities of states $i_0, i_1, \ldots, i_{k-1}$. The probabilities of the states satisfy the following equations as can be seen from the diagram.

$$\sum_{l=0}^{k-1} p_{i_l} + p_{j_l} = 1$$

$$p_{i_l} = p_j^l p_{i_0}$$

and

$$p_{j_l} = p_i^l p_{j_0} \qquad 0 \leqslant l < k$$

$$p_{i_0} = (1-p_j)\sum_{l=0}^{k-1} p_{i_l} + p_i p_{j_{k-1}}$$

$$p_{j_0} = (1-p_i)\sum_{l=0}^{k-1} p_{j_l} + p_j p_{i_{k-1}}$$

The solution of this system of equations is

$$p_{j_m} = \frac{p_j^k p_i^m}{p_j^k \sum_{l=0}^{k-1} p_i^l + p_i^k \sum_{l=0}^{k-1} p_j^l}$$

$$p_{i_m} = \frac{p_i^k p_j^m}{p_j^k \sum_{l=0}^{k-1} p_i^l + p_i^k \sum_{l=0}^{k-1} p_j^l},$$

as can be easily verified. Thus we have shown the following lemma:

LEMMA 2.1.1.

$$b_k(j,i) = \frac{p_j^k \sum_{l=0}^{k-1} p_i^l}{p_j^k \sum_{l=0}^{k-1} p_i^l + p_i^k \sum_{l=0}^{k-1} p_j^l}$$

Now let $F_k(\vec{p})$ denote the (asymptotic) values of the expected number of probes necessary to perform a search in a table ordered under the simple k heuristic with the move to front rule. Then;

$$F_k(\vec{p}) = \sum_{i=1}^{n} \sum_{j=1}^{i} \frac{p_i^k p_i \sum_{l=0}^{k-1} p_i^l + p_i^k p_j \sum_{l=0}^{k-1} p_j^l}{p_j^k \sum_{l=0}^{k-1} p_i^l + p_i^k \sum_{l=0}^{k-1} p_j^l}.$$

Noting that the average search cost under the optimal ordering is $Opt(\vec{p}) = \sum_{i=1}^{n} i p_i$ and that we would like to bound the ratio of these expressions, consider the ratio of the m^{th} terms of the two expressions which can be simplified as

$$\frac{F_k(\vec{p})}{Opt(\vec{p})} \leqslant \max_m \left\{ \frac{1}{m} \sum_{j=1}^{m} \frac{p_j^k \sum_{l=0}^{k-1} p_m^l + p_j p_m^{k-1} \sum_{l=0}^{k-1} p_j^l}{p_j^k \sum_{l=0}^{k-1} p_m^l + p_m^k \sum_{l=0}^{k-1} p_j^l} \right\}$$

This term is, of course, bounded by the maximum of the terms in the summation. In other words, it is bounded by the maximum of

$$\frac{p^k(1+q+\ldots q^{k-1}) + pq^{k-1}(1+p+\ldots+p^{k-1})}{p^k(1+q+\ldots+q^{k-1}) + q^k(1+p+\ldots+p^{k-1})},$$

subject to $0 \leqslant q \leqslant p \leqslant 1$.

$$= \frac{p^k + pq^{k-1}\frac{1+p+\ldots+p^{k-1}}{1+q+\ldots+q^{k-1}}}{p^k + q^k\frac{1+p+\ldots+p^{k-1}}{1+q+\ldots+q^{k-1}}} \leqslant \frac{p^k + pq^{k-1}k}{p^k + q^k k}$$

$$= \frac{(p/q)^k+(p/q)k}{(p/q)^k+k} = \frac{x^k+kx}{x^k+k} \qquad \text{where } x=p/q \geqslant 1$$

The maximum of this term occurs when x satisfies the equation

$$(1-k)x^k+kx^{k-1}+k = 0$$

and

$$x = 1+\frac{\ln k}{k}+o\left(\frac{\ln k}{k}\right).$$

To prove this last result, we substitute $x = 1+\frac{a\ln k}{k}$ in the above polynomial and after some routine derivations we find

$$(1-k)x^k+kx^{k-1}+k = k^a-ak^a\ln k+k+O((\ln k)^2k^{a-1}).$$

For any $a \geqslant 1$ the polynomial becomes negative for a sufficiently large k and similarly for $a<1$ the polynomial becomes positive. Consequently the root of the polynomial occurs for

$$x = 1+\frac{\ln k}{k}+o\left(\frac{\ln k}{k}\right).$$

THEOREM 2.1.2. *The ratio of the expected number of probes required to perform a linear search on a table ordered by the simple k heuristic with the move to front rule, $F_k(\vec{p})$, to the search cost under the optimal ordering is bounded by*

$$\frac{F_k(\vec{p})}{Opt(\vec{p})} \leqslant 1+O(\frac{\ln k}{k}).$$

Proof. Substitute the asymptotic value of the root in $\frac{x^k+kx}{x^k+k}$. In particular we note that

$$F_2(\vec{p}) \leqslant (1+(1+\sqrt{3})^{-1})Opt(\vec{p}) \approx 1.36602...Opt(\vec{p}),$$

$$F_3(\vec{p}) \leqslant 1.27388...Opt(\vec{p}),$$

and

$$F_4(\vec{p}) \leqslant 1.22788...Opt(\vec{p}). \quad \square$$

LEMMA 2.1.3. *For $k>1$ and $p_i \geqslant p_j$*

$$p_j(1+p_i+...+p_i^{k-1})(1+p_j+...+p_j^{k-2}) \leqslant p_i(1+p_j+...+p_j^{k-1})(1+p_i+...+p_i^{k-2}).$$

Proof. By induction on k. For $k=2$: $p_j(1+p_i) \leqslant p_i(1+p_j)$. Assume that the inequality holds for k i.e.

$$p_ja(b-p_j^{k-1}) \leqslant p_ib(a-p_i^{k-1}) \qquad (*)$$

where $a = 1+p_i+...+p_i^{k-1}$ and $b = 1+p_j+...+p_j^{k-1}$.

We also know that $p_j^k(1-p_i^k) \leqslant p_i^k(1-p_j^k)$ or

$$p_j^k(1-p_i)a \leqslant p_i^k(1-p_j)b \qquad (**)$$

Adding (*) and (**) we obtain

$$p_j ab - a p_i p_j^k \leqslant p_i ab - b p_i^k p_j$$

or

$$p_j b(a + p_i^k) \leqslant p_i(b + p_j^k)a \quad \square$$

COROLLARY 2.1.4. *For $k>1$ and $p_i \geqslant p_j$ then*

$$p_j(1-p_i^k)(1-p_j^{k-1}) \leqslant p_i(1-p_j^k)(1-p_i^{k-1})$$

THEOREM 2.1.5. *For $k>1$, $F_k(\vec{p}) \leqslant F_{k-1}(\vec{p})$. Furthermore, the inequality is strict for non-trivial distributions.*

Proof. For $i<j$ and therefore $p_i \geqslant p_j$, the following is valid:
By Corollary 2.1.4. $p_j(1-p_i^k)(1-p_j^{k-1}) \leqslant p_i(1-p_j^k)(1-p_i^{k-1})$ or

$$p_i^{k-1}p_j^k(1-p_i^k)(1-p_j^{k-1}) \leqslant p_i^k p_j^{k-1}(1-p_j^k)(1-p_i^{k-1})$$

and hence

$$p_j^{2k-1}\frac{1-p_i^k}{1-p_i}\frac{1-p_i^{k-1}}{1-p_i} + p_i^{k-1}p_j^k\frac{1-p_i^k}{1-p_i}\frac{1-p_j^{k-1}}{1-p_j}$$
$$\leqslant p_j^{2k-1}\frac{1-p_i^k}{1-p_i}\frac{1-p_i^{k-1}}{1-p_i} + p_i^k p_j^{k-1}\frac{1-p_j^k}{1-p_j}\frac{1-p_i^{k-1}}{1-p_i}$$

or

$$\frac{p_j^k\frac{1-p_i^k}{1-p_i}}{p_j^k\frac{1-p_i^k}{1-p_i}+p_i^k\frac{1-p_j^k}{1-p_j}} \leqslant \frac{p_j^{k-1}\frac{1-p_i^{k-1}}{1-p_i}}{p_j^{k-1}\frac{1-p_i^{k-1}}{1-p_i}+p_i^{k-1}\frac{1-p_j^{k-1}}{1-p_j}}$$

and therefore

$$b_k(j,i) \leqslant b_{k-1}(j,i)$$

The theorem follows from [5, Theorem 2]. $\square$

2.2. Batched k Heuristics. A slightly different approach is to view requests as (for purposes of reorganization) being batched into groups of k consecutive requests. A reordering permutation is then applied only if all k requests in a batch were for the same element. The effect of such schemes may seem to be equivalent to the simple k approach. However, if an element is accessed and subsequently not moved forward because of the access of some other (unknown) element, the original element has a lower probability of being moved forward than in the case of the basic k in a row scheme. Intuitively, these heuristics are better than their counterparts of simple k heuristics because these heuristics perform fewer changes.

For a batched k heuristic with the move to front rule, the probability that record j precedes record i ($b'(j,i)$) can be derived in an analogous manner to that for $b_k(j,i)$ (in section 2.1). Indeed the following Markov chain describes this process.

FIGURE II

Markov Chain for Batched k algorithm.

It can be easily verified from the above Markov chain that

$$b_k' = \frac{p_j^k}{p_i^k+p_j^k}$$

Intuitively, the effect of the batched k heuristics is to raise the probability of access of each record to the power k which after normalization makes the large probabilities larger and the small probabilities even smaller.

Define $F_k'(\vec{p})$ and $T_k'(\vec{p})$ for the batched schemes in the same way that $F_k(\vec{p})$ and $T_k(\vec{p})$ were defined for the simple k schemes.

THEOREM 2.2.1. *For $k>1$, $F_k'(\vec{p}) \leqslant F_k(\vec{p})$. The inequality is strict for all nontrivial distributions.*

Proof. For $i<j$ and thus $p_i \geqslant p_j$

$$p_j{}^{2k}\sum_{l=0}^{k-1} p_i^l + p_i^k p_j^k \sum_{l=0}^{k-1} p_j^l \leqslant p_i^k p_j^k \sum_{l=0}^{k-1} p_i^l + p_j{}^{2k}\sum_{l=0}^{k-1} p_j^l.$$

Thus

$$\frac{p_j^k}{p_i^k+p_j^k} \leqslant \frac{p_j^k \sum_{l=0}^{k-1} p_i^l}{p_j^k \sum_{l=0}^{k-1} p_i^l + p_i^k \sum_{l=0}^{k-1} p_j^l},$$

or

$$b_k'(j,i) \leqslant b_k(j,i) \qquad \text{for } i<j$$

The theorem follows from [5, Theorem 2]. □

THEOREM 2.2.2. *For $k>1$, $T_k'(\vec{p}) \leqslant T_{k-1}'(\vec{p})$. The inequality is strict for all non-trivial distributions.*

Proof. Similar to that of Theorem 2.2.1. □

LEMMA 2.2.3. *The polynomial $(k-1)x^k - kx^{k-1} - 1$ has a root at $x = 1+\frac{a}{k}+O(k^{-2})$ where a is the root of $(a-1)e^a = 1$.*

Proof. Let $x = 1+\frac{a}{k}+\frac{b}{k^2}+O(k^{-3})$. Then

$$\ln x = \frac{a}{k}+\frac{2b-a^2}{2k^2}+O(k^{-3}),$$

and

$$x^k = e^a(1+\frac{2b-a^2}{2k}+O(k^{-2})).$$

$$x^{k-1} = \frac{x^k}{x} = e^a(1+\frac{2b-a^2-2a}{2k}+O(k^{-2}))$$

Substituting x^k and x^{k-1} in the original polynomial, we get

$$ke^a(1+\frac{2b-a^2}{2k})-e^a(1+\frac{2b-a^2}{2k})-ke^a(1+\frac{2b-a^2-2a}{2k})-1+O(k^{-1}) = 0$$

or

$$-e^a+ae^a+1+O(k^{-1}) = 0.$$

Thus a solves the equation $e^a(a-1) = 1$, or $a = 1.27846....$ □

THEOREM 2.2.4. *The ratio of the expected number of probes required to perform a linear search on a table ordered by the batched k heuristic with the move to front rule, $F_k'(\vec{p})$, to the search cost under the optimal ordering is bounded by*

$$\frac{F_k'(\vec{p})}{Opt(\vec{p})} \leqslant 1+\frac{a-1}{k}+O(k^{-2})$$

where a is the solution of $e^a(a-1) = 1$ or $a = 1.27846....$

In particular, solving the polynomial exactly for k =2,3 and 4 we obtain

$$F_2'(\vec{p}) \leqslant \frac{1+\sqrt{2}}{2}Opt(\vec{p}) = 1.20710...Opt(\vec{p}),$$

$$F_3'(\vec{p}) \leqslant 1.11843...Opt(\vec{p}),$$

$$F_4'(\vec{p}) \leqslant 1.08302...Opt(\vec{p}).$$

Proof.

$$b'(j,i) = \frac{p_j^k}{p_i^k+p_j^k},$$

$$F_k'(\vec{p}) = \sum_{i=1}^{n} p_i\left(1+\sum_{i'=j}\frac{p_j^k}{p_j^k+p_i^k}\right),$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{i}\frac{p_ip_j^k+p_i^kp_j}{p_i^k+p_j^k}.$$

Noting that the average search cost under the optimal ordering is $Opt(\vec{p}) = \sum_{i=1}^{n} ip_i$ and considering the ratio of the m^{th} terms of the two expressions, we get

$$\frac{F_k'(\vec{p})}{Opt(\vec{p})} \leqslant \max_m\left\{\frac{1}{m}\sum_{j=1}^{m}\frac{p_j^k+p_m^{k-1}p_j}{p_j^k+p_m^k}\right\}.$$

This term is bounded by the maximum of the terms in the summation. In other words, it is bounded by the maximum of

$$\frac{p^k+q^{k-1}p}{p^k+q^k} \qquad 0\leqslant q\leqslant p\leqslant 1$$

or

$$\frac{x^k+x}{x^k+1} \qquad x \geqslant 1.$$

This ratio is maximized when x satisfies $(k-1)x^k - kx^{k-1} - 1 = 0$. By Lemma 2.2.3, the ratio is maximized when $x = 1+\frac{a}{k}+O(k^{-2})$. Thus,

$$\frac{x^k+x}{x^k+1} = 1+\frac{x-1}{x^k+1} \leqslant 1+\frac{\frac{a}{k}+O(k^{-2})}{e^a(1+O(k^{-1}))+1}$$

$$\leqslant 1+\frac{\frac{a}{k}}{e^a+1}+O(k^{-2}) = 1+\frac{a-1}{k}+O(k^{-2}). \quad \square$$

THEOREM 2.2.5. *Under the batched k heuristic with transposition rule the stationary probabilities obey*

$$\frac{Prob[R_{i_1}R_{i_2}\ldots R_{i_j}R_{i_{j+1}}\ldots R_{i_n}]}{Prob[R_{i_1}R_{i_2}\ldots R_{i_{j+1}}R_{i_j}\ldots R_{i_n}]} = \frac{p_{i_j}^k}{p_{i_{j+1}}^k}$$

Proof.

$$Prob[R_{i_1}\ldots R_{i_n}] =$$

$$(1-p_{i_2}^k-p_{i_3}^k-\ldots-p_{i_n}^k)Prob[R_{i_1}\ldots R_{i_n}] + \sum_{i \leqslant j < n} p_{i_j}^k Prob[R_{i_1}\ldots R_{i_{j+1}}R_{i_j}\ldots R_{i_n}]$$

With the equation stated in the theorem, this implies

$$Prob[R_{i_1}\ldots R_{i_n}] = Prob[R_{i_1}\ldots R_{i_n}]. \quad \square$$

COROLLARY 2.2.6.

$$\frac{Prob[R_1\ldots R_{i-1}R_i\ldots R_{i+l}\ldots R_n]}{Prob[R_1\ldots R_{i-1}R_{i+l}\ldots R_i\ldots R_n]} = \left(\frac{p_i^k}{p_{i+l}^k}\right)^l.$$

THEOREM 2.2.7. $T_k'(\vec{p}) \leqslant F_k'(\vec{p})$ *for all distributions, furthermore the inequality is strict for nontrivial distributions.*

Proof.

$$Prob[\text{record } j \text{ precedes record } i] = \frac{p_j^k}{p_i^k+p_j^k}$$

for batched k heuristic with move to front rule. For batched k heuristic with transposition rule we have

$$Prob[\text{record } j \text{ precedes record } i] = \sum_{k=0}^{n-2} \sum_{\alpha} Prob[\alpha]$$

where α is a configuration for which record j precedes record i, with exactly l items between them.

$$= \sum_{k=0}^{n-2} \left\{ \left(\frac{p_j^k}{p_i^k}\right)^{l+1} \sum_{\beta} Prob[\beta] \right\}$$

where β is a configuration identical to α except that record j and record i are interchanged.

$$\leqslant \left(\frac{p_j^k}{p_i^k}\right)(1-Prob\,[record\ j\ precedes\ record\ i])$$

Thus

$$Prob\,[record\ j\ precedes\ record\ i] \leqslant \frac{p_j^k}{p_i^k+p_j^k} \quad \square$$

2.3. A brief disgression to self-organizing binary search trees. In the case of the self-organizing linear list, the k heuristic applied to the move to front rule perform uniformly better than the move to front heuristic. But, in the case of self-organizing binary search trees [2], the batched k scheme applied to the move to root heuristic is not always better than the simple move to root heuristic. For example, if $p_1 = 0.4$, $p_2 = 0.3$ and $p_3 = 0.3$, the cost for move to root is 1.88286... and the cost for the batched 2 scheme applied to move to root is 1.88306... . The intuitive explanation for this fact is that under the batched scheme applied to move to root heuristic, the key with the largest probability of access is more inclined to become the root of the tree than in the case for the simple move to root. However, having the most frequently accessed key as the root of the tree does not necesarily produce the best tree. The trees are depicted in Figure III

FIGURE III

Three possible trees

Furthermore, although we are able to decrease the cost of the batched k scheme as k increases in the case of linear list, the cost of the tree obtained by the batched k scheme applied to the move to root heuristic will not necessarily decrease as k increases. This occurs, for example, in the three key tree noted above and also in the case in which n keys have equal probability of being accessed. In the latter example the move to root heuristic both with and without a batched scheme produce a tree with a cost of about $(2/\ln 2)\log n$. We note that this is the worst possible behaviour relative to the optimal solution for the simple move to root heuristic.

3. More on the Move to Front Heuristic. In this section our attention returns to memory-free heuristics and in particular to the move to front rule. We analyze the expected behaviour of this rule for a number of distributions and demonstrate what is apparently a rather tight upper bound on the ratio $F(\vec{p})/Opt(\vec{p})$ for a class of distributions. This leads to the observation that this ratio can be as large, and no larger than, $\pi/2 \approx 1.57$ for an interesting class of distributions. It also gives us some intuition as to the difficulty of closing the gap between the worst known case of this ratio, and the best known upper bound, 2. Finally an expression for arbitrary moments of the number of accesses required by the move to front heuristic under any distribution is given.

3.1. Analyses of the Cost of the Move to Front Rule under Several Distributions. Knuth [6] analyzed the expected cost of a search under the move to front rule. We present the results of such an analysis for this and several other interesting distributions. The most interesting observation is that the ratio $\frac{F(\vec{p})}{Opt(\vec{p})}$ for Lotka's Law is $\frac{\pi}{2}$ (H_n denotes the n^{th} harmonic number. $H_n^{(l)}$ denotes $\sum_{i=1}^{n} i^{-l}$). We are interested primarily in the case in which n, the number of elements in the list, tends to ∞ .)

(i) Zipf's Law. Under Zipf's law elements have accessing probabilities $p_i = \frac{1}{iH_n}$. The optimal ordering has a cost $Opt(\vec{p}) = n/H_n$. The simple move to front rule has a cost [6]:

$$F(\vec{p}) = 2\ln 2\frac{n}{H_n} - \frac{1}{2} + o(1)$$

In this case we can also compute

$$F_2'(\vec{p}) = \frac{1}{2} + \frac{1}{H_n}\sum_{j=1}^{n}\sum_{i=1}^{n}\frac{i}{i^2+j^2}$$

$$= \frac{1}{2} + \frac{1}{H_n}\sum_{j=1}^{n} j\left(-\frac{1}{2j^2} + \frac{\pi\coth(\pi j)}{2j} - \sum_{i=n+1}^{\infty}\frac{1}{i^2+j^2}\right)$$

$$= \frac{\pi}{2H_n}\left(\sum_{j=1}^{n}\coth(\pi j) - \sum_{j=1}^{n}\tan^{-1}(j/n)\right) + O(1) \approx \frac{n}{H_n}\left(\frac{\pi}{4} - \frac{\ln 2}{2}\right) \approx 1.13197...\frac{n}{H_n}.$$

(ii) For Lotka's Law the accessing probabilities are $p_i = (i^2 H_n^{(2)})^{-1}$, and the optimal arrangement has cost $Opt(\vec{p}) = H_n/H_n^{(2)}$.

$$F(\vec{p}) = \frac{1}{2} + \frac{1}{H_n^{(2)}}\sum_{j=1}^{n}\sum_{i=1}^{n}\frac{1}{i^2+j^2}$$

$$= \frac{1}{2} + \frac{1}{H_n^{(2)}}\sum_{j=1}^{n}\left(-\frac{1}{2j^2} + \frac{\pi\coth(\pi j)}{2j} - \sum_{i=n+1}^{\infty}\frac{1}{i^2+j^2}\right),$$

$$= \frac{1}{H_n^{(2)}}\sum_{j=1}^{n}\left(\frac{\pi\coth(\pi j)}{2j} - \frac{\tan^{-1}(j/n)}{j} + O(n^{-2})\right)$$

$$= \frac{3\ln n + 3\gamma + 6C_1}{\pi} - \frac{6\beta(2)}{\pi^2} + O\left(\frac{\ln n}{n}\right)$$

$$F(\vec{p}) = \frac{3}{\pi}\ln n - 0.00206339... + O\left(\frac{\ln n}{n}\right),$$

where $C_1 = -\sum_j \ln(1-e^{-2\pi j}) = 0.001872...$ and $\beta(2) = 0.91596...$ is Catalan's constant [1].

(iii) For the exponential distribution we consider that $n \to \infty$ and $p_i = (1-a)a^{i-1}$. Consequently $Opt(\vec{p}) = \frac{1}{1-a}$.

$$F(\vec{p}) = \frac{1}{2}+\sum_{j=1}^{\infty}\sum_{i=1}^{\infty}\frac{(1-a)a^i a^j}{a(a^i+a^j)} = \frac{1}{2}+\frac{(1-a)}{a}\left(\frac{a}{2(1-a)}+2\sum_{j=1}^{\infty}\sum_{k=1}^{\infty}\frac{a^{i+j}}{1+a^j}\right),$$

$$= 1+2\sum_{j=1}^{\infty}\frac{a^j}{1+a^j}.$$

Using the modified Euler-Maclaurin summation formula [4] we obtain

$$F(\vec{p}) = -\frac{2\ln 2}{\ln a}-\frac{1}{2}-\frac{\ln a}{24}-\frac{\ln^3 a}{2880}+O(\ln^5 a).$$

(iv) For the wedge distribution $p_i = \frac{2(n+1-i)}{(n+1)n}$ and $Opt(\vec{p}) = \frac{n}{3}+\frac{2}{3}$. It is straightforward to compute

$$F(\vec{p}) = \frac{4(1-\ln 2)}{3}n - H_n + \frac{5(1-\ln 2)}{3}+\frac{H_n}{n}+O(n^{-1}).$$

3.2. Upper Bounds on $\frac{F(\vec{p})}{Opt(\vec{p})}$ for a Class of Distributions. Rivest [8] has shown that the average cost of the move to front rule is at most twice that of the optimal ordering. It has been conjectured that this bound is not tight, and indeed the worst value known for this ratio is $\frac{\pi}{2}$ as seen in the preceding subsection. In this subsection we derive an upper bound for the ratio, $F(\vec{p})/Opt(\vec{p})$, for the class of distributions $p_i \sim i^{-\lambda}$. Observing that $Opt(\vec{p})$ is bounded by a constant when $\lambda>2$, we will see that over the distributions of the above form for which the average search time is not bounded, Lotka's Law is the distribution which maximizes the ratio $F(\vec{p})/Opt(\vec{p})$. Our analysis seems very good in the range $0\leqslant\lambda\leqslant 2$, but weakens above this range.

Case (i) $0\leqslant\lambda\leqslant 2$

$$\frac{F(\vec{p})}{Opt(\vec{p})} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{i}\frac{p_i p_j}{(p_i+p_j)}}{\sum_{i=1}^{n} i p_i}.$$

Taking the m^{th} term of the numerator and denominator we have

$$\frac{F(\vec{p})}{Opt(\vec{p})} \leqslant \frac{2\sum_{j=1}^{m}\frac{p_m p_j}{(p_m+p_j)}}{m p_m} = \frac{2}{m}\sum_{j=1}^{m}\frac{1}{1+p_m/p_j}$$

$$= \frac{2}{m}\sum_{j=1}^{m}\left(1-\frac{p_m}{p_j}+\left(\frac{p_m}{p_j}\right)^2-\ldots\right).$$

Substituting $(\frac{p_m}{p_j}) = (\frac{j}{m})^{\lambda}$ and doing some manipulation we can show that

$$\frac{F(\vec{p})}{Opt(\vec{p})} \leqslant \lim_{l\to\infty}\left[\frac{1}{\lambda}[\psi(l+\frac{1}{2\lambda})-\psi(\frac{1}{2\lambda})-\psi(l+\frac{\lambda+1}{2\lambda})+\psi(\frac{\lambda+1}{2\lambda})]\right],$$

where l ranges over the integers and $\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$ is the Psi function [1]. Computing this limit we obtain:

THEOREM 3.1. *Over the class of probability distributions of the form* $p_i \sim i^{-\lambda}$ *for* $0 \leqslant \lambda \leqslant 2$. *(Note that* $Opt(\vec{p})$ *is not bounded in this case).*

$$\frac{F(\vec{p})}{Opt(\vec{p})} \leqslant \frac{1}{\lambda}\left[\psi(\frac{\lambda+1}{2\lambda})-\psi(\frac{1}{2\lambda})\right].$$

We note that in the relevant range this is a monotonic increasing function of λ and

COROLLARY 3.2. *For probability distributions of the form* $p_i \sim i^{-\lambda}$ *such that* $Opt(\vec{p})$ *diverges:* $\frac{F(\vec{p})}{Opt(\vec{p})} \leqslant \frac{\pi}{2}$. *Equality is achieved when* $\lambda = 2$.

It is interesting to note a few other values for λ and the upper bound on the ratio

λ	Upper bound on Ratio
0 (Uniform)	1
$\frac{1}{2}$	$4(1-\ln 2) = 1.2274...$
1 (Zipf)	$2\ln 2 = 1.3863...$
$\frac{3}{2}$	$\frac{4}{3}(\frac{\pi}{\sqrt{3}}-\ln 2) = 1.4942...$
2 (Lotka)	$\frac{\pi}{2} = 1.5708...$

TABLE I

We note that these bounds are tight for $\lambda = 0,1$ and 2, which are the only values in the range for which we have a precise analysis.

Consider now the other case:

Case (ii) $\lambda > 2$

Our analysis is made easier if we let $n \to \infty$, then it can be shown for this range that

$$p_i = \frac{1}{\zeta(\lambda) i^{\lambda}}$$

(where ζ denotes the Riemann Zeta function). Thus

$$Opt(\vec{p}) = \sum_{i=1}^{\infty} i p_i = \frac{\zeta(\lambda-1)}{\zeta(\lambda)},$$

and

$$F(\vec{p}) = \frac{1}{2}+\sum_{j=1}^{\infty}\sum_{j=1}^{\infty}\frac{p_i p_j}{(p_i+p_j)}$$
$$= \left(\frac{2}{\zeta(\lambda)}\right)\sum_{j=1}^{\infty}\sum_{i=j}^{\infty}\frac{1}{(i^{\lambda}+j^{\lambda})}.$$

Asymptotic analysis using the Euler-Mclaurin summation formula shows that the fourth term is negative in the range $2\leqslant\lambda\leqslant 4$, hence taking the first three terms we find an upper bound:

$$F(\vec{p}) < \frac{\zeta(\lambda-1)}{\zeta(\lambda)}\left[\frac{2\pi}{\sin(\pi/\lambda)}-\frac{1}{\lambda}\left[\psi\left(\frac{\lambda+1}{2\lambda}\right)-\psi\left(\frac{1}{2\lambda}\right)\right]\right]+\frac{1}{2}$$
$$+\frac{\lambda\zeta(\lambda+1)}{48\zeta(\lambda)}+\frac{\lambda(\lambda^2-4)\zeta(\lambda+3)}{5760\zeta(\lambda)}$$

For $\lambda = 2.1$ this value is roughly 1.5. It decreases monotonically to $\lambda = 4$ where it is about 1.1. The bound appears to be reasonably good, although we have discarded terms and so it is definitely not tight.

For $\lambda\geqslant 4$ the summation that defines $F(\vec{p})$ becomes rapidly convergent and using simple arguments we can bound it by

$$F(\vec{p}) \leqslant 1+\frac{2}{\zeta(\lambda)(1+2^{\lambda})}+\frac{2}{\lambda-2}3^{2-\lambda}.$$
$$\leqslant 1.21981... \qquad \text{for } \lambda\geqslant 4$$

and equivalently since $Opt(\vec{p}) > 1$,

$$\frac{F(\vec{p})}{Opt(\vec{p})} \leqslant 1+\frac{2}{\zeta(\lambda)(1+2^{\lambda})}+\frac{2}{\lambda-2}3^{2-\lambda}.$$

3.3. Worst cases of the Move to Front rule. From the above discussion one wonders which is the worst possible case for the move to front rule. There are two interesting worst cases, one given by the distribution that maximizes $\frac{F(\vec{p})}{Opt(\vec{p})}$ and the second is given by the one that maximizes $F(\vec{p})-Opt(\vec{p})$. Let

$$\alpha_k = \sum_{j=1}^{n}\left(\frac{p_j}{p_k+p_j}\right)^2$$

then

THEOREM 3.3.1. *The probability distribution that maximizes the ratio follows*

$$\frac{\alpha_k-\alpha_1}{k-1} = \frac{F(\vec{p})}{2Opt(\vec{p})}.$$

Proof. By taking partial derivatives of $\frac{F(\vec{p})}{Opt(\vec{p})}$ with respect to p_k. □

THEOREM 3.3.2. *The probability distribution that maximizes the difference* $F(\vec{p})-Opt(\vec{p})$ *follows*

$$\frac{\alpha_k-\alpha_1}{k-1} = \frac{1}{2}.$$

Proof. By taking partial derivatives of $F(\vec{p})-Opt(\vec{p})$ with respect of p_k. □

Unfortunately we do not know an explicit form for the worst cases, although we can compute the distributions for small n. The following two graphs show the worst ratio and the worst difference for small values of n.

FIGURE IV
Worst Ratio and Worst difference for Move to Front Rule

3.4. Arbitrary Moments of the Move to Front Rule. A closed form for the average cost of the memory-free move to front heuristic is given by Rivest [8] and used extensively in this paper. Also of interest are higher moments of this value and of course the variance, which can be derived as follows.

Let B_{ij} be the random variable defined by

$$B_{ij} = \begin{cases} 1 & \text{if } record\ i\ precedes\ record\ j \\ 0 & otherwise \end{cases}$$

Then let $E[B_{ij}] = b(i,j) = \dfrac{p_i}{p_i+p_j}$

$$E[(accesses)^m] = E\left[\sum_{j=1}^{n} p_j\left(\sum_{i\neq j} B_{ij}+1\right)^m\right] = \sum_{j=1}^{n} p_j E\left[\left(\sum_{i\neq j} B_{ij}+1\right)^m\right].$$

Let $S_i(m) = \sum_{p\geqslant i}\begin{bmatrix} m \\ i \end{bmatrix}$. We derive

$$E[(accesses)^m] = \sum_{j=1}^{n} p_j\left[1+S_1(m)E\left[\sum_{i\neq j} B_{ij}\right]+S_2(m)E\left[\sum_{i,k\neq j} B_{ij}B_{kj}\right]+\ldots\right].$$

Finally using the expression for the probability of a given permutation [5] we conclude

THEOREM 3.3. *The* m^{th} *moment of the distribution is given by*

$$E[(accesses)^m] = 1+2S_1(m)\sum_{i<j}\frac{p_ip_j}{p_i+p_j}$$

$$+4S_2(m)\sum_{i<k<j}\frac{p_ip_kp_j}{p_i+p_k+p_j}\left(\frac{1}{p_j+p_k}+\frac{1}{p_i+p_j}+\frac{1}{p_i+p_k}\right)+\ldots$$

$$+2r!S_r(m)\sum_{i_1<i_2<\ldots<i_r<i_{r+1}}\frac{p_{i_1}p_{i_2}\cdots p_{i_r+1}}{p_{i_1}+p_{i_2}+\ldots+p_{i_{r+1}}}I(p_{i_1},p_{i_2},\ldots,p_{i_r+1}),$$

where

$$I(a_1,a_2,\ldots,a_l) = \sum\frac{1}{a_{i_1}+a_{i_2}}\frac{1}{a_{i_1}+a_{i_2}+a_{i_3}}\cdots\frac{1}{a_{i_1}+a_{i_2}+\ldots+a_{i_{l-1}}}$$

the summation being over all permutations $i_1 i_2 \ldots, i_{l-1}$ of the integers $1..l$.

In particular for $m = 1$,

$$F(\vec{p}) = 1+2\sum_{i<j} \frac{p_i p_j}{p_i+p_j},$$

and for $m = 2$,

$$E[(accesses)^2] = 1+6\sum_{i<j} \frac{p_i p_j}{p_i+p_j}+4 \sum_{i<k<j} \frac{p_i p_k p_j}{p_i+p_k+p_j} \left(\frac{1}{p_j+p_k}+\frac{1}{p_i+p_j}+\frac{1}{p_i+p_k} \right).$$

From this we can demostrate

COROLLARY 3.4. *The variance of the move to front heuristic is*

$$var(\vec{p}) = (2-F(\vec{p}))(F(\vec{p})-1) + 4 \sum_{i<k<j} \frac{p_i p_k p_j}{p_i+p_k+p_j} \left(\frac{1}{p_i+p_j}+\frac{1}{p_j+p_k}+\frac{1}{p_k+p_i} \right).$$

The form of this value is quite different from the expression for the variance given by McCabe [7].

4. Conclusion. We have demonstrated a technique for maintaining self-organizing linear files in near optimal order without significant memory requirements. Although the analysis given can probably be tightened, it is sufficient to demonstrate the value of the approach. The behaviour of the simple move to front heuristic has been analyzed for a class of distributions. It is shown that over all distributions in this class the ratio of the cost of the move to front scheme to that of optimal ordering is at most $\pi/2$, and that this bound can be achieved. We conjecture that this is the maximum value this ratio can achieve over any class of distributions whose optimal search cost is unbounded.

REFERENCES

[1] M. Abramowitz, and I.A. Stegun, *Handbook of Mathematical Functions*, Dover Publications, New York.

[2] B. Allen, and J.I. Munro, *Self-organizing search trees*, J. Assoc. Comput. Mach., 25, (1978), pp. 526-535.

[3] J.R. Bitner, *Heuristics that dynamically organize data structures*, this Journal, 8, (1979), pp. 82-110.

[4] G.H. Gonnet, *Notes on the derivation of asymptotic expressions from summations*, Information Processing Letters, 7(1978), pp. 165-169.

[5] W.J. Hendricks, *An account of self-organizing systems*, this Journal 5(1976), pp.715-723.

[6] D.E. Knuth, *The Art of Computer Programming, Searching and Sorting*, Vol. III, Addison-Wesley, Don Mills, Ont. 1973.

[7] J. McCabe, *On serial files with relocatable records*, Operations Res. 12 (1965), pp. 609-618.

[8] R. Rivest, *On self-organizing sequential search heuristics, Comm. ACM, 19, (1976), pp. 63-67.*

[9] A. Tanenbaum, *Simulations of dynamic sequential search algorithms*, Comm. ACM 21, (1978), pp. 790-791.

† This work was supported by the Natural Sciences and Engineering Research Council of Canada under grants A8237 and A3353.

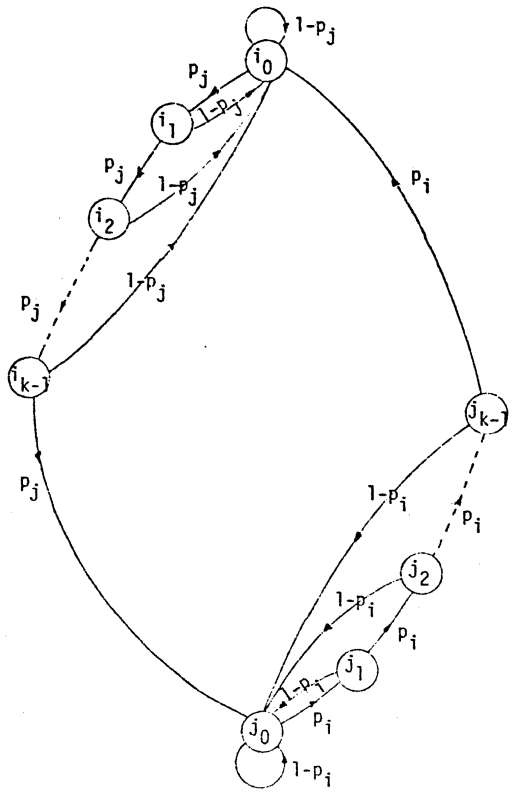


FIGURE I

Markov Chain for the simple k heuristic.

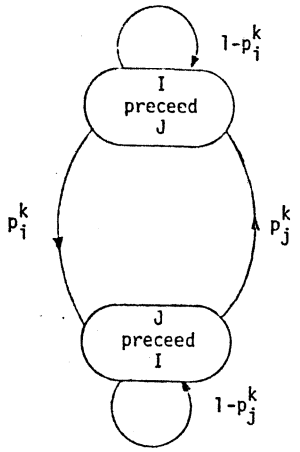
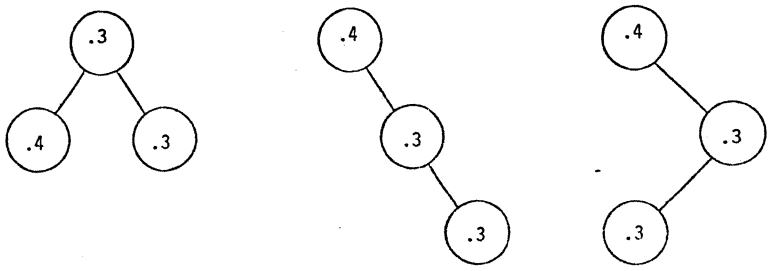


FIGURE II

Markov Chain for Batched k algorithm.



Optimal tree
Cost = 1.7

The two likeliest trees of the batched 2 applied to move to root heuristic.

FIGURE III

Three possible trees.

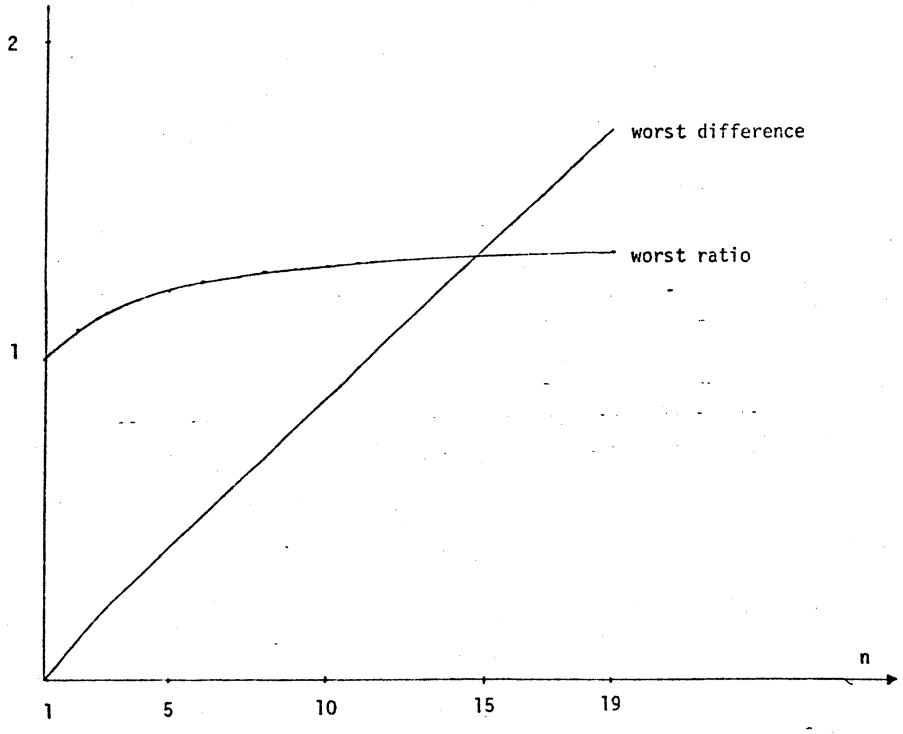


FIGURE IV

Worst Ratio and Worst difference for Move to Front Rule.