A New Approach to the Analysis of Linear Probing Schemes

HAIM MENDELSON

University of Rochester, Rochester, New York

AND

URI YECHIALI

Tel-Aviv University, Tel-Aviv, Israel

ABSTRACT. A new approach to the analysis of hash table performance is presented. This approach is based on a direct probabilistic analysis, where the underlying probabilities are derived by using the ballot theorem and its ramifications. The method is first applied to analyze the performance of the classical (cyclic) linear probing scheme, and the results are used to solve an optimal storage allocation problem. A scheme frequently used in practice where the table is linear rather than cyclic is then analyzed using the same methodology.

KEY WORDS AND PHRASES: hashing, collision resolution, linear probing scheme, cyclic probing scheme, ballot theorem

CR CATEGORIES: 3.72, 4.33, 5.3, 5.5

1. *Introduction*

A random-access table consists of N storage locations 1, 2, 3, . . . , N. Records are added to the table from time to time. Each record is identified by a unique key, w. A record with key w is hashed to storage location $h(w)$, where the hashing function $h(\cdot)$ is given (see [3, 5]). If the calculated location $h(w)$ is empty, the record is stored there. However, since $h(\cdot)$ is not one to one, location $h(w)$ may be occupied by another record. Such an occurrence is called a *collision.*

Several methods for collision resolution are known [1, 3, 6–8, 10]. In an "open" addressing system [8], there is a set of rules which determines, for each acceptable key w, a *probe sequence* of possible storage locations in which the corresponding record might be stored. The record is normally stored in the first location, $h(w)$, of the sequence. If that location is occupied, the second location in the sequence is tried, and so on, until an empty location is found.

The simplest open addressing scheme, which is known in the literature as *linear probing* [3, 8], scans the table sequentially (in a cyclic manner) until an empty location is found. The search for an empty location for storing a record with key w is performed along the probe sequence

$$(h(w), h(w) + 1, h(w) + 2, \ldots, N, 1, 2, \ldots, h(w) - 1).$$

This scheme has been suggested by Peterson [8] and analyzed by Konheim and Weiss [4] and by Knuth [3].

In this study we present a new approach to the analysis of the linear probing scheme.

Authors' addresses: H. Mendelson, Graduate School of Management, University of Rochester, Rochester, NY 14627; U. Yechiali, Department of Statistics, Tel-Aviv University, Ramat-Aviv, Tel-Aviv, Israel.

Our approach is based on the application of ballot theorems. An important feature of the analysis is that its methodology is independent of the circular symmetry of the addressing system. As a result, the method can be further used to analyze the performance of a linear table.

The structure of the paper is as follows. Section 2 contains a preliminary discussion of the ballot theorem and presents the renumbering technique and the "dual ballot theorem." The new approach is introduced in Section 3 and used to analyze the performance of a cyclic table (i.e., the classical linear probing scheme). A problem of optimal storage allocation is solved in Section 4. Section 5 concludes the paper with the analysis of a linear table.

2. *The Ballot Theorem*

Let $(X_1, X_2, \ldots, X_p)$ be a random vector, and let $S_n = \sum_{j=1}^{n} X_j$ $(n = 1, 2, 3, \ldots, p)$. The analyses made in this paper require a study of the distributions of $\max_{n=1,2,\ldots,p}\{S_n - n\}$ and $\max_{n=1,2,\ldots,p}\{n - S_n\}$, where $X_1, X_2, \ldots, X_p$ are interchangeable. This issue has been studied extensively in [9], where the classical ballot theorem was generalized and applied to various problems.

We say that the random variables $X_1, X_2, \ldots, X_p$ are interchangeable if all the $p!$ permutations of $X_1, X_2, \ldots, X_p$ have the same joint distribution. Throughout this section $(X_1, X_2, \ldots, X_p)$ is an arbitrary vector of interchangeable random variables taking on nonnegative integral values. Such a vector satisfies the classical ballot theorem.

THEOREM 1 (THE BALLOT THEOREM)

$$P\{S_n < n \text{ for } n = 1, 2, 3, \ldots, p \mid S_p = m\} = \left(1 - \frac{m}{p}\right)^{+},$$

where

$$(X)^{+} \equiv \max\{X, 0\}.$$

One of the numerous proofs of Theorem 1 (under weaker assumptions) is given in [9, p. 10].

The distribution of $\max_{n=1,2,3,\ldots,p}\{n - S_n\}$ is given by the following theorem, which is proved by Takács [9, p. 24].

THEOREM 2. *For* $k = 1, 2, 3, \ldots,$

$$P\{\max_{n=1,2,3,\ldots,p} \{n - S_n\} < k\} = 1 - \sum_{n=k}^{p} P\{S_n = n - k\} \cdot \frac{k}{n}.$$

In what follows we often make use of symmetry arguments based on the interchangeability of $X_1, X_2, X_3, \ldots, X_p$. The joint distribution of $(X_{j_1}, X_{j_2}, X_{j_3}, \ldots, X_{j_n})$ is independent of the specific selection of $j_1, j_2, j_3, \ldots, j_n$ $(n \leq p)$, as long as they are all distinct. Since $(X_{j_1}, X_{j_2}, X_{j_3}, \ldots, X_{j_n}) \sim (X_1, X_2, X_3, \ldots, X_n)$, the probability of each event which is defined in terms of $X_{j_1}, X_{j_2}, X_{j_3}, \ldots, X_{j_n}$ may be computed by replacing X_{j_i} $(i = 1, 2, 3, \ldots, n)$ with X_i. In the sequel we refer to this probability-preserving transformation as a *renumbering*.

By renumbering $(X_1, X_2, X_3, \ldots, X_n)$ in a reverse order, we obtain the dual sequence $(X_1^*, X_2^*, X_3^*, \ldots, X_n^*)$, where $X_j^* = X_{n+1-j}$ $(j = 1, 2, 3, \ldots, n)$. Use of this simple transformation yields Theorem 3, which may be considered as the dual of the classical ballot theorem.

THEOREM 3 (DUAL BALLOT THEOREM)

$$P\{S_n \geq n \text{ for } n = 1, 2, 3, \ldots, m \mid S_{m+1} = m\} = (m + 1)^{-1}.$$

PROOF. Let $(X_1^*, X_2^*, X_3^*, \ldots, X_{m+1}^*)$ be the dual sequence of $(X_1, X_2, X_3, \ldots, X_{m+1})$,

and let $S_n^* = \sum_{j=1}^{n} X_j^* = S_{m+1} - S_{m+1-n}$ for $n = 1, 2, 3, \ldots, m+1$ ($S_0^* \equiv 0$). Use of Theorem 1 for the dual sequence yields

$$\begin{aligned}(m+1)^{-1} &= P\{S_n^* < n \text{ for } n = 1, 2, 3, \ldots, m \mid S_{m+1}^* = m\} \\ &= P\{S_{m+1-n} \geq m+1-n \text{ for } n = 1, 2, 3, \ldots, m \mid S_{m+1} = m\} \\ &= P\{S_n \geq n \text{ for } n = 1, 2, 3, \ldots, m \mid S_{m+1} = m\},\end{aligned}$$

since $S_n^* < n$ is equivalent to $S_{m+1-n} > m - n$ or $S_{m+1-n} \geq m + 1 - n$ when $S_{m+1}^* = S_{m+1} = m$. Q.E.D.

3. *Analysis of Cyclic Linear Probing*

The classical linear probing scheme has been studied by Konheim and Weiss [4] and by Knuth [3]. We introduce our new approach by performing an analysis of this scheme. Under the linear probing scheme the table is scanned in a cyclic manner along the probe sequence $(h(w), h(w)+1, \ldots, N, 1, 2, \ldots, h(w)-1)$. Nevertheless, the underlying ideas of our analysis do not collapse in the absence of the circular symmetry of the above addressing system. This fact will become more evident in Section 5 where we perform the analysis of a linear table.

Consider a circular table containing k records, where $k < N$. Let X_j be a random variable denoting the number of records mapped by $h(\cdot)$ to the jth location ($j = 1, 2, 3, \ldots, N$). The distribution of the random vector $(X_1, X_2, \ldots, X_N)$ is multinomial with parameters k (= number of trials) and $((1/N), (1/N), \ldots, (1/N))$ (= probabilities for success). We further define $S_0 = 0$, and for $p = 1, 2, 3, \ldots, N$ we let $S_p = \sum_{n=1}^{p} X_n$ be the cumulative number of records hashed to the first p locations. For the analysis we extend the numbering of storage locations in a cyclic manner, i.e., if $j \equiv j' \pmod N$, then both j and j' represent the same location. We also define $S_{N+j} = S_N + S_j$ and $S_{-j} = S_{N-j} - S_N$, so that for all $j \leq j'$, the cumulative number of records hashed to locations $(j+1, j+2, \ldots, j')$ is $S_{j'} - S_j$.

The records fill some portions of the (extended) table, which we call *strings*. We say that a string of length m ($m = 1, 2, 3, \ldots, k$) starts at location j, if (see Figure 1):

(i) location $(j-1)$ is empty;
(ii) locations $j, j+1, j+2, \ldots, j+m-1$ are all occupied;
(iii) location $j+m$ is empty.

This event will be denoted by $E_{j,m}(k)$, and its probability by $P_{j,m}(k)$. Once the underlying probabilities $P_{j,m}(k)$ are known, it is possible to analyze the performance of the table.

It is easy to see that $E_{j,m}(k)$ occurs if and only if the following events occur simultaneously:

$$\begin{aligned}&(E_1) \quad S_{j+m} - S_{j-1} = m; \\ &(E_2) \quad S_{j-1+n} - S_{j-1} \geq n \quad \text{for} \quad n = 1, 2, 3, \ldots, m; \\ &(E_3) \quad S_{j-1} - S_{j-1-n} < n \quad \text{for} \quad n = 1, 2, 3, \ldots, N-m-1.\end{aligned}$$

E_1 implies that exactly m records are hashed to storage locations $(j, j+1, j+2, \ldots, j+m)$. E_2 guarantees that all the locations within the string are occupied, whereas E_3 implies that location $(j-1)$ remains empty. An essential point to note is that given E_1, the events E_2 and E_3 are conditionally independent. This follows since the distribution of records among locations $(j, j+1, j+2, \ldots, j+m)$ is independent of the internal record distribution among the remaining locations. Hence,

$$P_{j,m}(k) = P(E_1) \cdot P(E_2 \mid E_1) \cdot P(E_3 \mid E_1). \tag{1}$$

So far we have not invoked the specific form of the addressing system. Focusing on the cyclic linear probing scheme, the circular symmetry of the system implies that $P_{j,m}(k)$ is independent of j. Hence, it may be assumed that $j = 1$, and so,

$$P(E_1) = P(S_{m+1} = m) = \binom{k}{m} \left(\frac{m+1}{N}\right)^m \left(\frac{N-m-1}{N}\right)^{k-m} \tag{2}$$

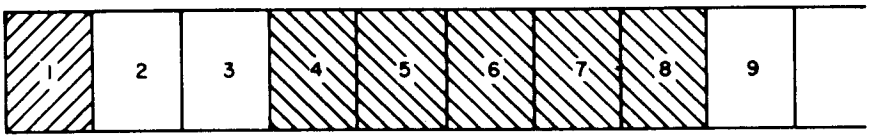

FIG. 1. A string of length $m = 5$ that starts at location $j = 4$. The shaded areas represent occupied locations.

Next we compute

$$P(E_2 | E_1) = P\{S_n \geq n \text{ for } n = 1, 2, 3, \ldots, m | S_{m+1} = m\} = (m + 1)^{-1}, \tag{3}$$

as follows from the dual ballot theorem (Theorem 3).

Last, by using the interchangeability of the X_j's, we obtain

$$\begin{aligned} P(E_3 | E_1) &= P\{S_{-n} < n \text{ for } n = 1, 2, 3, \ldots, N - m - 1 | E_1\} \\ &= P\{S_n < n \text{ for } n = 1, 2, 3, \ldots, N - m - 1 | S_{N-m-1} = k - m\} \\ &= 1 - \frac{k - m}{N - m - 1} = \frac{N - k - 1}{N - m - 1}, \end{aligned}$$

where the probability is evaluated by using the ballot theorem (Theorem 1).

Combining our results we obtain

THEOREM 4. *For* $k = 1, 2, 3, \ldots, N - 1$; $j = 1, 2, 3, \ldots, N$; $m = 1, 2, 3, \ldots, k$,

$$P_{j,m}(k) = \frac{1}{N^k} \binom{k}{m} \cdot (m + 1)^{m-1}(N - m - 1)^{k-m} \cdot \frac{N - k - 1}{N - m - 1}. \tag{4}$$

The same result has been obtained by Konheim and Weiss [4] and by Knuth [3].

The result of Theorem 4 may now be applied to analyze the performance of a circular table with N locations, k of which are occupied. The relevant performance measures are

(i) $D(k)$, the expected number of extra probes needed for an unsuccessful search;
(ii) $C(k)$, the expected number of extra probes needed for a successful retrieval.

In both cases, we do not count the initial probe at location $h(w)$.

It is easily seen that the operation of record addition is in fact an unsuccessful search which is followed by storing the new record, whereas an update in place is composed of a successful search and a store operation. Furthermore, we have

$$C(k) = \frac{1}{k} \sum_{n=0}^{k-1} D(n), \tag{5}$$

when each of the k records in the table is equally likely to be retrieved.

The derivation of $D(k)$ and $C(k)$ from $P_{j,m}(k)$ has been performed by Knuth [3, p. 530]. The result can be formulated as follows.

THEOREM 5

$$D(k) = \frac{1}{2} \sum_{j=0}^{k} \binom{k}{j} \frac{(j + 1)!}{N^j} - \frac{1}{2} \quad \textit{for} \quad k = 0, 1, 2, \ldots, N - 1, \tag{6}$$

and

$$C(k) = \frac{1}{2} \sum_{j=0}^{k-1} \binom{k - 1}{j} \frac{j!}{N^j} - \frac{1}{2} \quad \textit{for} \quad k = 1, 2, 3, \ldots, N. \tag{7}$$

In the next section we apply these results to solve a storage allocation problem.

4. *An Optimization Application*

Consider the problem of allocating storage for a hash table, where the number of records to be registered in the table is not known a priori. For example, storage is allocated for the symbol table of a compiler before the source code is examined; hence the number of

records to be stored in the table is a random variable. Assume that this number has a Poisson distribution with mean a; each record in the table is to be retrieved q times on the average, and the storage-allocation cost of each entry is equivalent to the cost of s probes. The objective is to minimize the total expected cost subject to the constraint that the table does not overflow with probability of at least $1 - \alpha$ ($\alpha \ll 1$).

Let N be the number of entries allocated for the table. We express the resulting costs in terms of the expected total number of probes. The storage-allocation cost is converted to the equivalent number of probes, which is $s \cdot N$. The cost of inserting the $(n+1)$st record into the table ($n = 0, 1, 2, \ldots$) is $D(n)$ extra probes. The cost of q retrievals of this record is $q \cdot D(n)$, since each retrieval follows the same probe sequence scanned when the record was inserted. Hence the expected conditional total cost, given that k records have been registered in the table, is

$$s \cdot N + (q+1) \cdot \sum_{n=0}^{k-1} D(n) = s \cdot N + (q+1) \cdot k \cdot C(k),$$

which follows from (5). The requirement $\alpha \ll 1$ implies that the total expected cost may be approximated by

$$f(N) \equiv s \cdot N + (q+1) \cdot \sum_{k=0}^{\infty} k \cdot C(k) \cdot e^{-a} \frac{a^k}{k!}.$$

Using expression (7) for $C(k)$, we derive

$$\begin{aligned} f(N) &= s \cdot N - \frac{a(q+1)}{2} + \frac{e^{-a}(q+1)}{2} \cdot \sum_{k=0}^{\infty} \frac{k \cdot a^k}{k!} \cdot \sum_{j=0}^{k-1} \binom{k-1}{j} \frac{j!}{N^j} \\ &= s \cdot N - \frac{a(q+1)}{2} + \frac{e^{-a}(q+1)}{2} \cdot \sum_{j=0}^{\infty} \frac{1}{N^j} \cdot \sum_{k=j+1}^{\infty} \frac{a^k}{(k-1-j)!} \\ &= s \cdot N - \frac{a(q+1)}{2} + \frac{a(q+1)}{2} \cdot \sum_{j=0}^{\infty} \left(\frac{a}{N}\right)^j \\ &= s \cdot N + \frac{1}{2} \cdot (q+1) \cdot \frac{a^2}{(N-a)}. \end{aligned}$$

Now $f(x) = sx + \frac{1}{2}(q+1)a^2/(x-a)$ is a convex function which is minimized at

$$x^* = a\left(1 + \sqrt{\frac{q+1}{2s}}\right), \tag{8}$$

so N_1, the unconstrained optimal value of N, is either $[x^*]$ or $[x^*] + 1$.

Next, the requirement that the probability of table overflow is bounded by α is equivalent to $N \geq N_2$, where

$$N_2 = \min\left\{n \mid \sum_{k=n+1}^{\infty} e^{-a} \frac{a^k}{k!} < \alpha\right\}.$$

The constrained optimal value of N is therefore

$$N^* = \max\{N_1, N_2\}.$$

For example, if each record is retrieved $q = 9$ times on the average, and $s = 20$, we obtain from (8) that $x^* = 1.5a$. When $a \geq 100$ (which is quite ordinary), x^* exceeds the mean a by at least five standard deviations; hence the probability of table overflow is practically zero, and $N^* = 1.5a$.

5. *Analysis of a Linear Table*

The classical (cyclic) linear probing scheme discussed in the previous sections may be replaced in real-life applications by a *linear table*. The linear table consists of M storage locations $1, 2, 3, \ldots, M$, where the hashing function $h(\cdot)$ maps arriving records only to the

subset $\{1, 2, 3, \ldots, N\}$, and the last $M - N$ locations form an "overflow area." When a record with key w is to be added to the table, it is stored in the first empty location found along the probe sequence $(h(w), h(w) + 1, h(w) + 2, \ldots, N, N + 1, \ldots, M)$. This addressing scheme has the advantage that if the table gets exhausted, it may simply be augmented with additional "overflow" entries, whereas other hashing schemes (e.g., cyclic linear probing) require rehashing of the records in the table [2]. Other minor advantages are a reduction in the number of page faults in a virtual-storage environment (due to the linear scanning scheme) and a somewhat simpler programming.

On the other hand, a linear table might become exhausted before it is completely full. That is, the search for an empty location might be terminated without finding one even when the number of records in the table, k, is lower than the maximum capacity, M. Thus the evaluation of the linear table depends heavily on the probability $F(k)$ that the table is exhausted when k records are present. We proceed with the derivation of this performance measure by applying the ballot theorem.

For the analysis, we add an imaginary location, the $(M + 1)$st, at the end of the table. We note that the table is exhausted if and only if this imaginary location is occupied. Thus $1 - F(k)$ is the probability that the $(M + 1)$st location remains empty. This happens if and only if $S_N - S_n < M + 1 - n$ for $n = 0, 1, 2, \ldots, N - 1$, where S_n is the total number of records hashed to the first n locations. Since $S_N = k$, we obtain, for $k = 0, 1, 2, \ldots, M$,

$$\begin{aligned} 1 - F(k) &= P\{n - S_n < M + 1 - k \text{ for } n = 0, 1, 2, \ldots, N - 1 \mid S_N = k\} \\ &= P\{\max_{n=1,2,3,\ldots,N-1} (n - S_n) < M + 1 - k \mid S_N = k\}. \end{aligned}$$

Using Theorem 2 we obtain

$$\begin{aligned} F(k) &= \sum_{n=M+1-k}^{N-1} \frac{M + 1 - k}{n} P\{S_n = n - (M + 1 - k) \mid S_N = k\} \\ &= \frac{M + 1 - k}{N^k} \sum_{n=M+1-k}^{N-1} \binom{k}{M + 1 - n} n^{n-1-(M+1-k)}(N - n)^{M+1-n}. \end{aligned} \tag{9}$$

In Figure 2 we illustrate the function $-\log_{10} F(k)$ for a linear table with $N = 50$, $M = 55$. Note that $F(k)$ is below 0.01 when the occupancy of the table does not exceed 60 percent.

Next we consider the search for an empty location in a linear table which contains k records. Let w be the key, which determines the initial probe, and let $U^{(k)}$ denote the number of additional probes needed for the search. Our purpose is to derive $D(k) = E[U^{(k)}]$. Since the random variable $U^{(k)}$ is well defined only when the system is not exhausted, we assume that M is large enough so that the probability $F(k)$ is negligible. Otherwise $D(k)$ depends on the exact way an exhausted table is handled. Our results correspond to the case where the table may be extended as required, so $D(k)$ is independent of M, and the contribution of the exhausted case to the total expected cost is obtained via $F(k)$.

Obviously, the definitions of $E_{j,m}(k)$ and $P_{j,m}(k)$ apply to the linear case as well. We now express the performance measure $D(k)$ in terms of the probabilities $P_{j,m}(k)$. Assume that the key w is hashed to storage location $i = h(w)$ $(i = 1, 2, 3, \ldots, N)$. If this location is empty, the search is completed and the number of additional probes is $U^{(k)} = 0$. Otherwise storage location i belongs to a string which starts at some location j $(j \le i)$. The number of additional probes is $U^{(k)} = r$ if and only if the string is $r + i - j$ locations long, where $r = 1, 2, 3, \ldots, k - (i - j)$. Hence

$$P\{U^{(k)} = r \mid h(w) = i\} = \sum_{j=1}^{i} P_{j,r+i-j}(k) \qquad \text{for} \quad r = 1, 2, 3, \ldots, k - (i - j).$$

It follows that

$$D(k) = \sum_{i=1}^{N} \frac{1}{N} \sum_{\substack{r=1 \\ r+i-j \le k}}^{k} r \sum_{j=1}^{i} P_{j,r+i-j}(k) = \frac{1}{N} \sum_{j=1}^{N} \sum_{n=1}^{k} P_{j,n}(k) \cdot \sum_{r=\max\{j+n-N,1\}}^{n} r.$$

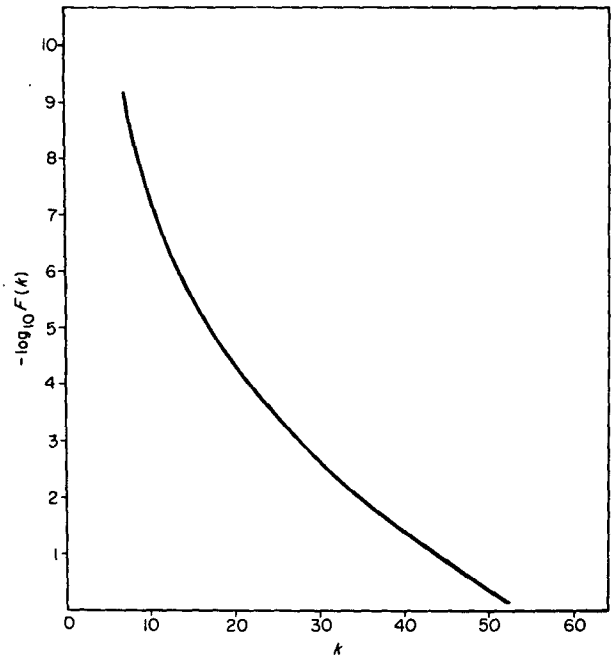

FIG. 2. $-\log_{10} F(k)$ for a linear table with $N = 50$, $M = 55$.

Partitioning the summation over n into two cases, we obtain

THEOREM 6. *The expected number of additional probes in an unsuccessful search is given by*

$$D(k) = \frac{1}{2N} \sum_{\substack{j=1 \\ j+n \le N}}^{N} \sum_{n=1}^{k} n(n+1) P_{j,n}(k) + \frac{1}{2N} \sum_{\substack{j=1 \\ j+n > N}}^{N} \sum_{n=1}^{k} (j + 2n - N)(N - j + 1) P_{j,n}(k). \tag{10}$$

It remains to derive the underlying probabilities $P_{j,m}(k)$ for the linear table. For $n = 1, 2, 3, \ldots$, we define $S_{N+n} = S_N$ and $S_{-n} = S_0 = 0$. Then $E_{j,m}(k) = E_1 \cap E_2 \cap E_3$, as defined in Section 3. Again, given E_1, the events E_2 and E_3 are conditionally independent; hence eq. (1) for $P_{j,m}(k)$ holds in this case too.

The formulas for $P_{j,m}(k)$ are given by

THEOREM 7. *For $j = 1, 2, 3, \ldots, N$; $m = 1, 2, 3, \ldots, k$, we have*

(i) if $j + m \le N$,

$$P_{j,m}(k) = \frac{1}{N^k} \binom{k}{m} (m+1)^{m-1} \cdot (N - m - 1)^{k-m} \cdot \frac{N - k - 1}{N - m - 1} + \frac{1}{N^k} \binom{k}{m} (m+1)^{m-1} \sum_{n=j}^{k-m} (n - j + 1) \binom{k-m}{n} (j-1)^{n-1} (N - m - j)^{k-m-n};$$

(ii) if $N < j + m \le M$,

$$P_{j,m}(k) = \frac{1}{N^k} \binom{k}{m} (j-1)^{k-m} \left(1 - \frac{k-m}{j-1}\right)^{+} \cdot \left[(N + 1 - j)^m - \sum_{n=1}^{N-j} \frac{1}{n} \binom{m}{n-1} n^{n-1} (N + 1 - j - n)^{m-(n-1)} \right],$$

where we define $0^0 \equiv 1$, and $0/0 \equiv 0$.

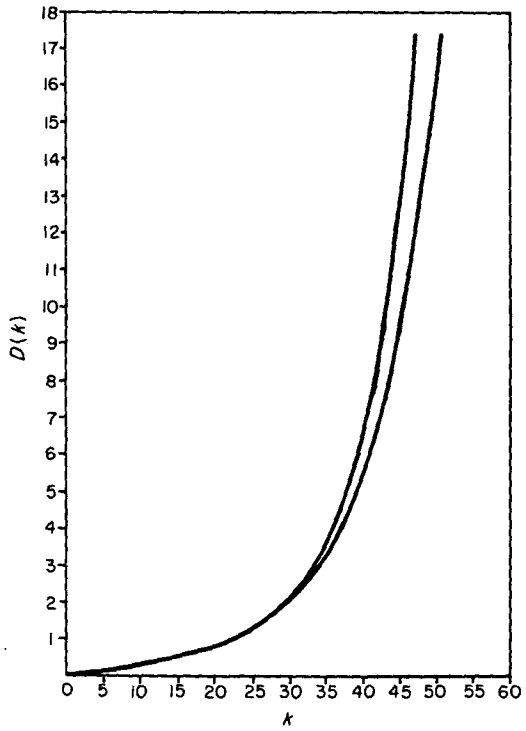

FIG. 3. $D(k)$ for a linear and a circular table with $N = 50$. The upper graph corresponds to the circular table.

PROOF

(i) If $j + m \leq N$, then $P(E_1)$ and $P(E_2 | E_1)$ are given by (2) and (3), respectively. Now,

$$P(E_3 | E_1) = P\{S_{j-1} - S_{j-1-n} < n \text{ for } n = 1, 2, 3, \ldots, j-1 \,|\, S_{j+m} - S_{j-1} = m\}.$$

Given that $S_{j+m} - S_{j-1} = m$, $S_{j-1} \sim B(k-m, (j-1)/(N-m-1))$. By conditioning on S_{j-1} we obtain

$$P(E_3 | E_1) = \sum_{r=0}^{\min\{j-1,\, k-m\}} P\{S_{j-1} - S_{j-1-n} < n \text{ for } n = 1, 2, 3, \ldots, j-1 \,|\, S_{j-1} = r\} \cdot \binom{k-m}{r} \left(\frac{j-1}{N-m-1}\right)^r \left(1 - \frac{j-1}{N-m-1}\right)^{k-m-r}.$$

When $j = 1$, obviously $P(E_3 | E_1) = 1$. For $j = 2, 3, 4, \ldots, N-m$, we obtain, by renumbering storage locations $(1, 2, 3, \ldots, j-1)$ in a reverse order and using the ballot theorem,

$$P\{S_{j-1} - S_{j-1-n} < n \text{ for } n = 1, 2, 3, \ldots, j-1 \,|\, S_{j-1} = r\} = 1 - \frac{r}{j-1} \quad \text{for} \quad r = 0, 1, 2, \ldots, j-1.$$

Hence, noting that $\sum_{r=0}^{\min\{j-1,\, k-m\}} = \sum_{r=0}^{k-m} - \sum_{r=j}^{k-m}$, we have

$$P(E_3 | E_1) = 1 - \frac{k-m}{N-m-1} + \frac{1}{(N-m-1)^{k-m}} \cdot \sum_{r=j}^{k-m} (r-j+1) \cdot \binom{k-m}{r} (j-1)^{r-1} (N-m-j)^{k-m-r}. \tag{11}$$

Combining (2), (3), and (11) proves the case where $j + m \leq N$.

(ii) Next consider the case where $N < j + m \leq M$. Now $P(E_3 | E_1) = 1 -$

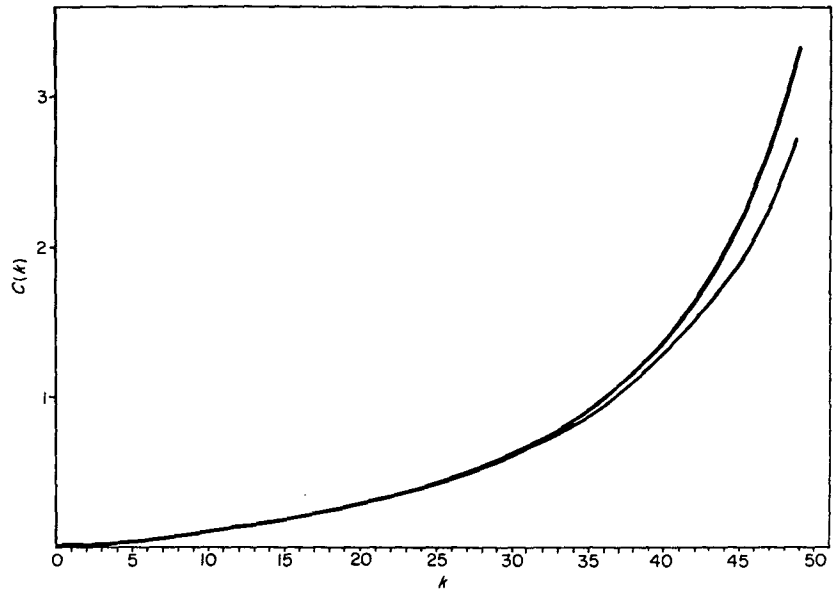

FIG. 4. $C(k)$ for a linear and a circular table with $N = 50$. The upper graph corresponds to the circular table.

$(k - m)/(j - 1)$ for $j > k - m + 1$, and $P(E_3 | E_1) = 0$ for $j \leq k - m + 1$. By using Theorem 2 we obtain, for $j = 1, 2, 3, \ldots, N - 1$,

$$P(E_2 | E_1) = P\{\max_{n=1,2,3,\ldots,N-j} (n - S_n) < 1 \mid S_{N+1-j} = m\}$$
$$= 1 - \sum_{n=1}^{N-j} \frac{1}{n} \binom{m}{n-1} \left(\frac{n}{N+1-j}\right)^{n-1} \left(1 - \frac{n}{N+1-j}\right)^{m-n+1}$$

This result is also valid for $j = N$, since then $P(E_2 | E_1) = 1$. Q.E.D.

COROLLARY. *When $k < j + m \leq N$, $P_{j,m}(k)$ is given by expression (4) of the circular case.*

The significance of this corollary is that off the edges of the table, the behavior of the system is similar to that of the circular table.

We conclude with graphic representations (Figures 3 and 4) of $D(k)$ and $C(k) = (1/k) \sum_{n=0}^{k-1} D(n)$. In these figures we compare the performance of a linear table with $N = 50$ to that of a circular table with the same value of N. Obviously, $D(K)$ (hence also $C(k)$) is lower in the linear case. This does not necessarily imply that the linear geometry is more efficient, since $F(k)$ also affects the final outcome. Note that while $D(k)$ increases up to $(N - 1)/2$ (in the circular case), $C(k)$ remains comparatively low.

REFERENCES

1. BLAKE, I.F., AND KONHEIM, A.G. Big buckets are (are not) better! *J. ACM 24*, 4 (Oct. 1977), 591–606.
2. CARTER, B. The reallocation of hash-coded tables. *Comm. ACM 16*, (Jan. 1973), 11–14.
3. KNUTH, D.E. *The Art of Computer Programming, Vol 3.* Addison-Wesley, Reading, Mass., 1973, Sec. 6.4, pp. 506–518.
4. KONHEIM, A.G., AND WEISS, B. An occupancy discipline and applications. *SIAM J. Appl. Math. 14* (Nov. 1966), 1266–1274.
5. LUM, V.Y., YUEN, P.S.T., AND DODD, M. Key-to-address transform techniques: a fundamental performance study on large existing formatted files. *Comm. ACM 14*, 4 (Apr. 1971), 228–239.

6. MENDELSON, H., AND YECHIALI, U. Performance measures for ordered lists in random access files. *J. ACM 26*, 4 (Oct. 1979), 654–667.
7. MORRIS, R. Scatter storage techniques. *Comm. ACM 11*, 1 (Jan. 1968), 38–44.
8. PETERSON, W.W. Addressing for random-access storage. *IBM J. Res. Devel. 1* (1957), 130–146.
9. TAKÁCS, L. *Combinatorial Methods in the Theory of Stochastic Processes.* John Wiley and Sons, New York, 1967.
10. VAN DER POOL, J.A. Optimum storage allocation for initial loading of a file. *IBM J. Res. Devel. 16* (1972), 579–586.

RECEIVED DECEMBER 1978; REVISED JUNE 1979; ACCEPTED AUGUST 1979.