ON THE OPTIMALITY OF RECTANGULAR HASHING SCHEMES FOR ANSWERING BASIC QUERIES FROM A RELATION WITH INDEPENDENT ATTRIBUTES

by

Azad Bolour

Memorandum No. ERL-M535

18 August 1975

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

ON THE OPTIMALITY OF RECTANGULAR HASHING SCHEMES FOR ANSWERING BASIC QUERIES FROM A RELATION WITH INDEPENDENT ATTRIBUTES †

by

Azad Bolour

Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory,
University of California, Berkeley, California 94720

ABSTRACT

This paper is concerned with the problem of finding efficient hashing schemes for answering basic queries (queries specified by a conjunction of attribute-value equalities), in a multi-attribute file residing in a secondary storage device.

A model problem is examined in which the n attributes in a record occur independently of one another, and for which the probability that a query specifies values for a particular subset of the attributes is given.

For this model it is shown that in many cases, rectangular (or multiple key) hashing schemes - hashing schemes that partition an n-dimensional attribute-value space into a regular lattice of rectangular sets in n-space - have near minimal average page access, in the class of all balanced hash functions from an attribute-value space, onto a given number of pages of equal size, in a secondary storage device.

† Research supported in part by the Naval Electronic Systems Command contracts N00039-71-C-0255/N0039-75-C-0034 and in part by the National Science Foundation grant GK-10656x.

I. INTRODUCTION

1. Overview

Consider an n-dimensional file (relation) F, consisting of records of the form $(x_1,x_2,...,x_n)$ chosen from a space of all possible records $K = K_1x...xK_n$, and suppose that F is to be stored in a number (say p) of pages of equal size in a secondary storage device. This paper is concerned with the problem of finding efficient hashing schemes for answering basic queries from such a file. A basic query, otherwise known as a partial-match query, specifies particular values for some subset of the attributes (components) of the records in a given file, and requires for its answer all records in the file possessing the specified attribute values. A hash function, on the other hand, is a mapping from the space K of all possible records, onto the p given pages, assigning to each record the page wherein it is to be stored in case it belongs to the file under consideration. We shall henceforth concern ourselves only with balanced hash functions, which informally speaking, have the property that they distribute the records in a typical file F evenly among the p given pages.[†] Notice that a hash function can be characterized by the way it partitions the record space into parts each of which corresponds to a given page.

One measure of the difficulty of answering a query Q from a file organized according to a hash function f, is the number of pages that <u>must</u> be examined if one is to give a complete answer to Q. Of course we shall assume that the algorithm for answering Q does not have any prior information about the contents of the file F, so that, neglecting page overflow, the

[†]This informal statement suffices for the present introductory discussion. All formal definitions, including that of a balanced hash function, appear in the next section.

number of pages that must be examined to answer Q depends only on Q and on the particular hash function being used. Thus, given a probability measure on the set of basic queries, the average number of pages that must be accessed to answer a basic query from a file organized according to a hash function f, may be taken as an appropriate overall measure of the performance of the filing scheme defined by f.

Simply stated then, the problem is to design a hash function minimizing such an average page access.

The general problem of file organization for efficient retrieval of records selected using several keys (attributes), or combinations of keys, has received considerable attention in the literature during the past decade [1,3,4,7,9,12-17]. An excellent summary of some of the work in this field may be found in Knuth [11, Section 6.5]. The problem of finding hash functions with minimal average page access for multi-attribute queries was apparently first discussed in Welch [16]. Welch considers a file F with binary valued attributes, and assumes that the records in F are uniformly distributed in a record space, and that all basic queries are equally likely. His so-called "bucketing problem" then requires a partition of the record space into p parts of equal size, minimizing the average page access for the induced (balanced) hash function. Rivest [14] has furnished an elegant solution to the above problem for a general discrete record space,

$$K = K_1 x \ldots x K_n, \quad K_i = \{1,2,\ldots,u_i\}, \quad i \leq i \leq n.$$

His approach relies on the simple observation that the average number of pages accessed to answer a query is minimized, if the average access to each individual page is minimized. Accordingly he defines an optimal bucket as a part of the record space K, with the property that the average number of queries needing that part for their answer, is a minimum over all parts having the same size. He is then able to characterize the general shape of such optimal buckets. In fact it turns out that in many cases very simple geometrical configurations, e.g. n-dimensional cubes, achieve the above minimum. One can then say, for example, that if it is possible to partition the record space K into p cubical parts, then the induced hash function onto p pages is optimal: it achieves minimum average page access.

Evidently however, uniformity assumptions made on the occurrence of records and queries in order to obtain the above results, restrict to some extent their applicability. This paper represents an initial attempt at removing some of these assumptions. Specifically we consider a continuous n-dimensional record space, $K = K_1 x \ldots x K_n$, $K_i \subseteq R$, $1 \leq i \leq n$, and assume that the occurrence of records in a given file is governed by a probability density $P_K(\underline{x})$ on K, which is independent in the n attributes. We then assume that for each subset c of the attributes, the probability ω_c that a query specifies the attributes in c is given but arbitrary. Also among queries specifying a given subset, say $\{i_1, \ldots, i_j\}$, of the attributes, we assume that the probability that a query specifies the values $v_{i_1}, \ldots, v_{i_j}$, is identical to the probability that a record attains these values.

The problem again is to find a balanced hash function minimizing the average page access for answering basic queries, given the parameters $P_K(\underline{x})$, and ω_c, $c \subseteq \{1, \ldots, n\}$.

Here, one can similarly define an optimal bucket and attempt to characterize its shape. The principal result of this paper then is that in many cases an n-dimensional rectangle has an optimal shape for the above problem. It follows that a partition of the record space K into parts of the form,

$$I^1_{j_1} \times I^2_{j_2} \times \ldots \times I^n_{j_n}, \quad i \leq j_i \leq N_i, \quad 1 \leq i \leq n,$$

with a suitable choice of the integers N_i, $1 \leq i \leq n$, and of the component parts I_{j_i}, $1 \leq j_i \leq N_i$, $1 \leq i \leq n$, often induces a near optimal balanced hash function on K.

For obvious reasons we have chosen to call hash functions of the above kind "rectangular hash functions" in this paper. Such hash functions were originally introduced in the literature by Rothnie and Lozano in [15], where they are referred to as "multiple key hash functions," and where their behaviour in answering single attribute queries is investigated in detail.

2. Records, Files, Queries, and Hash Functions

Basic Definitions and Assumptions

i) Definition 1

By a record we mean an n-tuple of the form,

$$(x_1,x_2,\ldots,x_n) \quad x_i \in K_i = [a_i,b_i] \subseteq R, \; 1 \leq i \leq n.$$

The components $x_1,\ldots,x_n$, of a record are also known as attributes or keys, and the record space

$$K = K_1 \times \cdots \times K_n,$$

will also be referred to as the attribute-value space or the key space.

Definition 2

A file or a relation is a set of records.

We assume that the file A to be stored can be time varying, but has approximately constant size $|A|$.

Definition 3

A basic query or a partial match query is a request for all records in a file having specified values for a given subset of the attributes. A basic query can be represented by a set of attribute-value equalities. For example,

$$x_1 = 2,\ x_3 = 5,\ x_{10} = 0$$

is a representation of a basic query whose answer is the set:

$$\{r \in A \mid r = (x_1, \dots, x_n) \wedge x_1 = 2 \wedge x_3 = 5 \wedge x_{10} = 0\}\ .$$

Definition 4

Consider a subset $c = \{i_1,\dots,i_j\}$ of the set of attributes $\{1,2,\dots,n\}$. A query of type c is a basic query in which the set of attributes specified is c. For example the query,

$$x_1 = 2,\ x_3 = 5$$

is of type $\{1,3\}$.

ii) A Model for the Occurrence of Records in a File

We shall assume in this paper a probabilistic model for the generation of records in a file A. This, of course, is not to say that records in a real world file are normally generated according to a known probability density, but that in many situations the occurrence of records in a file may be reasonably approximated by a probabilistic model.

In particular we shall assume that the records in A are generated independently of one another and randomly according to a given continuous probability density $P_K(\underline{x})$ on the attribute-value space K, which we shall refer to as the data probability density.

In what follows we shall only be concerned with the case where the data probability density is independent in the attributes, i.e. where,

$$P_K(\underline{x}) = P_{K_1 x \cdots x K_n}(x_1,\ldots,x_n) = P_{K_1}(x_1) x \ldots x \, P_{K_n}(x_n).$$

This will simplify the analysis considerably, and may be a legitimate assumption for an attribute-value space in which the attributes are intrinsically unrelated. An example may be,

last name x city of residence x make of car

in a Department of Motor Vehicles (DMV) file.

iii) Assumptions Concerning the Probability of Queries

The problem we would like to be able to solve is as follows. Given the data probability density $P_K(\underline{x})$, and a probability density on the set of basic queries, find a hash function having minimal average page access in answering basic queries.

However, in the absence of further assumptions about the query probabilities, the problem seems intractable mathematically and is perhaps overspecified from a practical point of view, as it is unrealistic to expect detailed estimates of query probabilities that are very reliable. The following assumption goes considerably beyond the assumption of uniformity of the query probabilities, but does not require detailed estimation of the latter.

We let the probability density that a query of type $c = \{i_1,\ldots,i_j\}$ specifies the values $v_{i_1},\ldots,v_{i_j}$, be given by a function of the form,

$$q_c(v_{i_1},\ldots,v_{i_j}) = \omega_c P_{K_{i_1}}(v_{i_1}) \times \ldots \times P_{K_{i_j}}(v_{i_j}). \tag{1}$$

This is perhaps the most reasonable first order assumption that one may make about query probabilities. In the first place it assumes that the probability, ω_c, that a query is of type c is given. (Notice that one can expect to arrive at a reasonable estimate of ω_c by monitoring incoming queries.) Within each query type, however, it assumes that the denser the data is in some neighborhood, the more frequent are the queries in that same neighborhood. This means for example, that among queries specifying a last name in the DMV file of the last section, queries using the name Smith occur proportionally more frequently than those using the name Goldstein.

iv) Balanced Hash Functions

Treating the records in a file A as indecomposable objects that are to be stored one by one, let us assume that it takes b bits to store a representation of each. Let us also assume that for the purpose of storing the records in A, we provided with $p = |A|/s$ primary pages (together with a few overflow pages) of size sb bits each in a secondary storage device.

A balanced hash function into p pages is then a function from the attribute-value space K onto the set of pages $\{1,\cdots,p\}$, for which a set S_i of points mapping into a given page i, $(1 \leq i \leq p)$, has probability 1/p (according to the probability density $P_K(\underline{x})$).
Since the sets S_i, $1 \leq i \leq p$, partition the attribute-value space K, we shall also regard a balanced hash function as a partition of K into p parts of equal probability, with the interpretation that records whose

attribute values are in the kth part will be stored in the kth page (overflow being negligible and handled by chaining).

The requirement of balancedness, of course, stems from the assumptions that the records are generated according to the probability density $P_K(\underline{x})$, and that the pages have equal size.

3. Problem Statement

In answering queries we assume that no auxiliary information is available about the set of records presently in storage. The retrieval algorithm is then very simple:

given a query,

$$x_{i_1} = v_{i_1},\ x_{i_2} = v_{i_2}, \ldots,\ x_{i_j} = v_{i_j}$$

access the kth page if and only if the kth part (i.e. the part of the key space associated with the kth page) contains a point $\underline{t} = (t_1,\ldots,t_n)$ such that

$$t_{i_1} = v_{i_1},\ldots,t_{i_j} = v_{i_j}.$$

Now for a set $X \subseteq R^n$ and a subset of coordinates

$$c = \{i_1,\ldots,i_j\} \subseteq \{1,\ldots,n\},$$

let the projection of X on the cartesian product of the coordinates in c be denoted by $\pi_c(X)$:

$$\pi_c(X) = \{(v_{i_1},\ldots,v_{i_j}) \mid \exists \underline{t} = (t_1,\ldots,t_n) \in X,$$

$$t_{i_1} = v_{i_1},\ t_{i_2} = v_{i_2},\ldots,t_{i_j} = v_{i_j}\}.$$

It is then easy to see that the kth page is needed for a query

$$x_{i_1} = v_{i_1}, \dots, x_{i_j} = v_{i_j}$$

$$\text{of type} \qquad c = (i_1, \dots, i_j) \quad ,$$

if and only if $(v_{i_1}, \dots, v_{i_j}) \in \pi_c(X_k)$,

where X_k is the part of the key space corresponding to the kth page.

Hence using the query probabilities $q_c(v_{i_1}, \dots, v_{i_j})$, $c \subseteq \{1, \dots, n\}$, given in (1), the average number of times the kth page is needed to answer a basic query is,

$$\sum_{\substack{c = \{i_1, \dots, i_j\} \\ \subseteq \{1, \dots, n\}}} \int_{\pi_c(X_k)} q_c(t_{i_1}, \dots, t_{i_j}) dt_{i_1} \dots dt_{i_j} = \sum_{c \subseteq \{1, \dots, n\}} \omega_c P(\pi_c(X_k))$$

where for $c = \{i_1, i_2 \dots, i_j\}$, $P(\pi_c(X))$ is the probability of the set $\pi_c(X)$ induced by the data probability density $P_K(\underline{x})$:

$$P(\pi_c(X)) = \int_{\pi_c(X)} P_{k_{i_1}}(t_{i_1}) \cdots P_{k_{i_j}}(t_{i_j}) dt_{i_1} \cdots dt_{i_j} .$$

Therefore, letting $\bar{a}$ be the average number of pages accessed in order to answer a basic query, we have, by the additive property of averages,

$$\bar{a} = \sum_{k=1}^{p} \sum_{c \subseteq \{1, \dots, n\}} \omega_c P(\pi_c(X_k)) \cdot$$

The problem of finding an optimal hashing scheme for our model, can now be stated as follows.

Given probabilities or weights ω_c corresponding to each subset c of the attributes $\{1,\ldots,n\}$, find a partition of the attribute value space K, into p parts $X_1,\ldots,X_p$, such that,

$$P(X_k) = \int_{X_k} P_K(\underline{x})dx_1 \cdots dx_n = \frac{1}{p} \tag{2}$$

and

$$\sum_{k=1}^{p} \sum_{c \subseteq \{1,\ldots,n\}} \omega_c P(\pi_c(X_k))$$

is minimized.

4. A Canonical Problem: Uniform Data on the Unit n-Cube

Consider the problem (2) of the last section with the additional condition that the attribute values have a uniform probability density on the unit n-cube: $P_K(\underline{x}) = 1$ for $\underline{x} \in [0,1]^n = K$. It can be restated as follows:

Find a partition of $[0,1]^n$ into p parts $X_1,\ldots,X_p$, such that,

$$|X_k| = 1/p$$

and

$$\bar{a} = \sum_{k=1}^{p} \sum_{c \in \{1,\ldots,n\}} \omega_c |\pi_c(X_k)| \quad \text{is minimized,} \tag{3}$$

where for a set $T \subset R^j$, $|T|$ denotes the size or the volume of T in R^j.

We are now going to show that by suitable 'stretching' of the coordinates, any problem of the form (2) can be reduced to a corresponding 'uniform' problem of type (3).

Lemma 1

There is a one to one correspondence between the feasible solutions to the problem (2), and the feasible solutions to the related problem (3), in such a way that the corresponding solutions achieve identical values for their respective objective functions.

Outline of Proof

Without loss of generality assume $P_{K_i}(x_i) \neq 0$ for all $x_i \in K_i = [a_i, b_i]$, $1 \leq i \leq n$.

The desired one to one correspondence between partitions of K and those of the unit n-cube, is induced by the following bijection:

$$g: \ K_1 x \ldots x \ K_n \rightarrow [0,1]^n \ ,$$

$$g(x_1, \cdots, x_n) = (g_1(x_1), \cdots, g_n(x_n))$$

where $$g_i(x_i) = \int_{a_1}^{x_i} P_{K_i}(t)dt \ , \ 1 \leq i \leq n.$$

It is now trivial to show that g induces a bijection from partitions of K onto partitions of $[0,1]^n$, which bijection satisfies the condition of the lemma.

Thus, for the purpose of finding optimal partitions of a rectangular attribute-value space K, with a data probability density that is independent in the n attributes, it is sufficient to consider the problem (3) above.

5. Solution Strategy

Although our proof techniques are different, we follow a strategy identical to that used by Rivest [14], in tackling the above problem.

Let
$$a_p^* = \inf_{\substack{|X| = 1/p \\ X \subseteq [0,1]^n}} \sum_{c \subseteq \{1,\ldots,n\}} \omega_c |\pi_c(X)| \tag{4}$$

and let us designate parts X^* of $[0,1]^n$ which achieve this infimum, as optimal parts. It then follows from (3) that for any balanced partition of $[0,1]^n$, the average page access $\bar{a}$ is bounded below by the quantity $p\, a_p^*$,

$$\bar{a} \geq p\, a_p^* . \tag{5}$$

This lower bound is achievable if and only if the attribute-value space $[0,1]^n$ can be covered by disjoint optimal parts.

As will be seen in section 2, Rivest's proof of the optimality of cubical parts for the case where all basic queries are equally likely, does not carry over to the present more general case. To tackle the present problem we consider a slightly different lower bound, which may in some cases be weaker than pa_p^* (and therefore of little use), but turns out to be equal to pa_p^* in many cases.

In particular let

$$\hat{a}_p = \inf_{\substack{X \subset R^n \\ |X| = 1/p}} \sum_{c \subseteq \{1,2,\ldots n\}} \omega_c |\pi_c(X)| \tag{6}$$

and designate parts achieving this infimum, as optimal parts of R^n. Then since

$$\hat{a}_p \leq a_p^*,$$

we see that the quantity $p\hat{a}_p$ is also a lower bound on the average page access, which is tight if and only if,

a) an optimal part of R^n fits within the bounds of the key space $[0,1]^n$, i.e. $\hat{a}_p = a_p^*$,

and b) the key space $[0,1]^n$ can be covered by disjoint optimal parts.

In section 2 we show that while searching for optimal parts of R^n, it is sufficient to restrict ones attention to rectangular solids in R^n, and hence the problem of finding such parts reduces to a simple constrained optimization problem.

Our computational results presented in section 3, indicate, that in a great many cases, optimal rectangles in R^n do fit within the bounds of the unit n-cube, and one can in fact partition the unit n-cube into approximately optimal parts, obtaining a hashing scheme with approximately minimal retrieval time as measured by the average number of pages accessed to answer basic queries.

II. The Minimal Projection Property of Rectangular Sets in R^n

1. Motivation

To find optimal parts of R^n and hence the lower bound $p\hat{a}_p$ of the last section on the average page access, we have to solve the following optimization problem.

Given probabilities ω_c, for each $c \subseteq \{1,2,\ldots,n\}$,

find $X \subset R^n$

to minimize $$\sum_{c \subseteq \{1,2,\ldots n\}} \omega_c |\pi_c(X)| \tag{7}$$

such that $|X| = 1/p$,

where $\pi_c(X)$ is the projection of X on the cartesian product of the coordinates in c, and for a set $T \subset R^j$, $|T|$ denotes the volume of T in R^j.

In this section we are going to prove that the above minimum, if it exists, is achievable by a rectangular set in R^n, that is, a set of the form,

$$X = I_1 \times I_2 \times \ldots \times I_n \qquad I_i \subset R \quad 1 \leq i \leq n.$$

The crucial step in this proof is to regard the average projection

$\sum_{c \subseteq \{1,\ldots,n\}} \omega_c |\pi_c(X)|$, of a set X on subsets of coordinates, as a "generalized perimeter" of X. We are then in a position to utilize proof techniques employed in the solution of the so-called isoperimetric problem [10, Chapter 2]: the problem of maximizing the volume of a set in R^n given its perimeter, (or equivalently the problem minimizing the

the perimeter of a set given its volume). One such proof technique due to Steiner [2, Chapter 1] follows.

The problem is to find a plane figure of a given perimeter having maximal area. To begin with it is trivial to verify that a plane figure that is not convex cannot have maximal area among all figures with the same perimeter.

Now given a convex figure A in R^2, let ℓ be a straight line that divides A into two parts A_1 and A_2, in such a way that the perimeter of A is halved. Without loss of generality let $|A_1| \geq |A_2|$.
One can then replace A_2 by the reflection of A_1 on the line ℓ, whereby one obtains a set of equal perimeter but possibly larger area. (Figure 1).

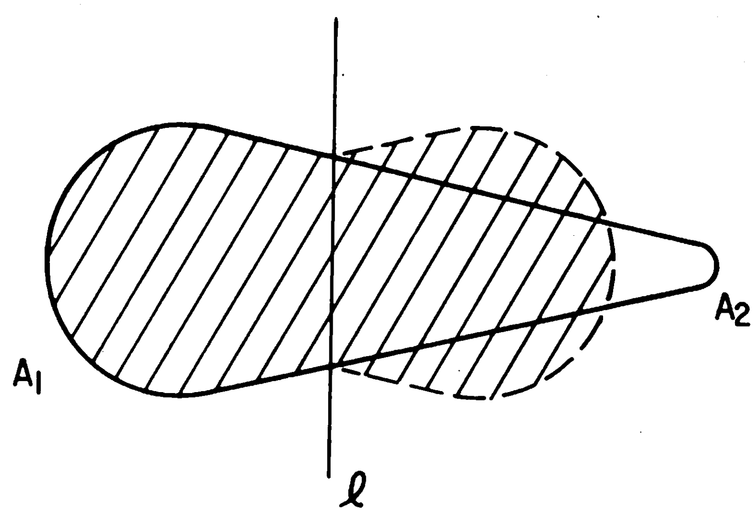

Figure 1

A subsequent construction shows that if the original figure were not a circle, then one can modify the derived symmetric figure, to obtain a figure of larger area but the same perimeter. Hence the maximum area of a figure of a given perimeter, if it exists, is attained by a circle.

The elegant use of symmetry in the above proof can be adapted to solve the problem of minimizing the average projection of a set of size s in R^2, as follows.

Let x_1 and x_2 represent the two coordinates in R^2. Given a set $X \subset R^2$, construct m lines parallel to the x_2 axis in such a way that they divide X into m+1 subsets $X_0, X_1, \ldots, X_m$ of equal size,

$$|X_i| = \frac{1}{m+1} |X| \cdot$$

(See figure 2, for the case m=1, where the construction is almost identical to Steiner's reflection.)

Let X_{i^*} have minimum projection, π_1^*, on the x_1 axis,

$$\pi_1^* = |\pi_1(X_{i^*})| \leq |\pi_1(X_i)|, \; 0 \leq i \leq m. \tag{8}$$

Now for each k, $0 \leq k \leq m$, construct a copy $\overline{X}_k$ of X_{i^*}, by translating the latter a distance $k\pi_1^*$ along the x_1 axis,

$$\overline{X}_k = \{(x_1 + k\pi_1^*, x_2) \mid (x_1, x_2) \in X_{i^*}\}.$$

Notice that $|\overline{X}_i \cap \overline{X}_j| = 0$ for $i \neq j$, and let $X_m^* = \bigcup_0^m \overline{X}_k$.

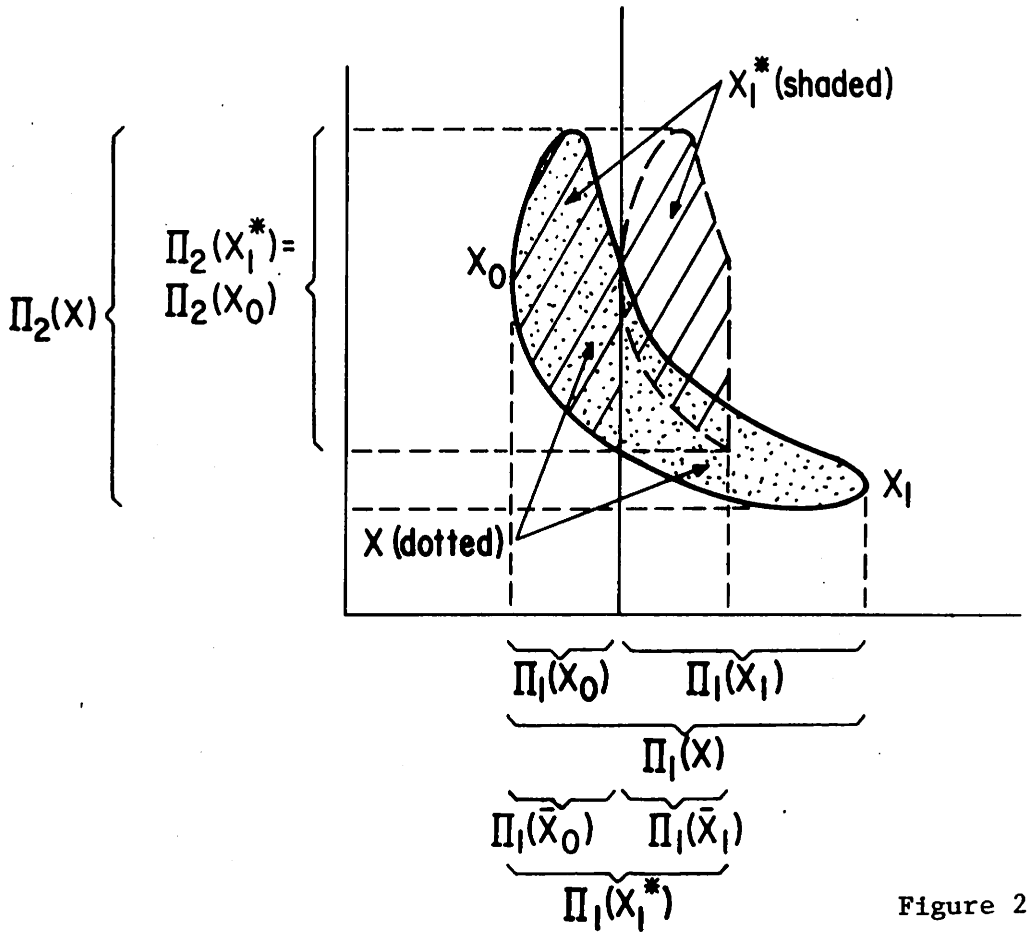

Figure 2

The set X_m^* satisfies:

1. $|X_m^*| = (m+1)|X_{i^*}| = |X|$

2. $|\pi_1(X_m^*)| = (m+1)|\pi_1(X_{i^*})|$

$$\leq \sum_0^m |\pi_1(X_i)| = |\pi_1(X)|$$

(the inequality follows from (8))

and 3. $|\pi_2(X_m^*)| = |\pi_2(X_{i^*})|$

$$\leq |\pi_2(\bigcup_0^m X_i)| = |\pi_2(X)|.$$

Hence X_m^* has the same area as X, but its projection on either coordinate is no larger than that of X. For large m, X_m^* is approximately rectangular and it is reasonable that as $m \to \infty$, X_m^* tends to a rectangle, and hence that the average projection of a plane figure of a given size must be minimized by a rectangle.

The same type of argument works in higher dimensions, the only difference being that $\pi_1(X_i)$ is everywhere replaced by,

$$\sum_{\substack{\text{all subsets } c_1 \text{ of} \\ \{1,\ldots,n\} \text{ containing } 1}} \omega_{c_1} |\pi_{c_1}(X_i)|$$

and similarly $\pi_2(X_i)$ is replaced by the corresponding summation over all subsets of coordinates not containing the first.

In an n-dimensional situation, however, it is again reasonable that as $m \to \infty$, the sets X_m^* approach a set of the form

$$[a,b] \times X^{n-1} \qquad X^{n-1} \subset R^{n-1},$$

in which case by repeating the argument for the remaining dimensions, one obtains a rectangular solid in R^n.

The above arguments of course, though intuitively appealing, are non-rigorous. The next section contains the details of a more rigorous statement of the same ideas.

2. The Minimal Projection Property of Rectangular Sets

Theorem 1

If the problem (7),

$$\text{minimize} \sum_{c \subseteq \{1,\dots,n\}} \omega_c |\pi_c(X)| \qquad (\omega_c \geq 0)$$

such that $X \subset R^n$, $|X| = 1/p$, has an optimal solution, then it has a rectangular optimal solution, $X^* = I_1 x \dots x I_n$, $I_i \subset R$, $1 \leq i \leq n$.

To prove the theorem we shall first have to develop some notation.

Let $R^n = R_1 \times R_2 \times \dots \times R_n$ where R_i, $1 \leq i \leq n$, are n copies of the real line.

Let c_0 denote a subset of the set $\{2,3,\dots,n\}$ of coordinates, and given c_0 let $c_1 = c_0 \cup \{1\}$.

Then given sets,

$$X_n \subset R_1 \times R_2 \times \dots \times R_n$$

$$X_{n-1} \subset R_2 \times \dots \times R_n$$

we let

1. $X_n(t) = \{(x_2,\dots, x_n) | (t,x_2,\dots, x_n) \in X_n\}$

2. $$\|X_{n-1}\|_0 = \sum_{c_0 \subseteq \{2,\dots,n\}} \omega_{c_0} |\pi_{c_0}(X_{n-1})|$$

   where by definition we let

   $$|\pi_\phi(X_{n-1})| = \begin{cases} 1 & X_{n-1} \neq \phi \\ 0 & X_{n-1} = \phi \end{cases}$$

3. $$\|X_{n-1}\|_1 = \sum_{\substack{c_1 = c_0 \cup \{1\} \\ c_0 \subseteq \{2,\dots,n\}}} \omega_{c_1} |\pi_{c_0}(X_{n-1})|$$

and 4. $\|X_n\| = \sum_{c \subseteq \{1,\ldots,n\}} \omega_c |\pi_c(X_n)|$.

Lemma 2

Let X be a subset of R^n such that $|X| = 1/p$. Then there exists a rectangular set, $Y = I_1 \times \ldots \times I_n$, $I_i \subset R$, $1 \leq i \leq n$, such that,

$$|Y| = 1/p$$

and

$$\|Y\| \leq \|X\|.$$

Proof

Let X be a subset of R^n, and without loss of generality assume there is an interval $I = \{x | a \leq (<) x \leq (<) b\} \subseteq R$, such that $|X(t)| \neq 0$ if and only if $t \in I$.

In case $X(t) \neq \phi$ for some $t \notin I$, replace X by

$$\bigcup_{t \in I} \{t\} \times X(t),$$

obtaining a set of equal volume but possibly smaller average projection.

Now

$$\|X\| = \sum_{c \subseteq \{1,\ldots,n\}} \omega_c |\pi_c(X)|$$

$$= \sum_{c_0 \subseteq \{2,\ldots,n\}} \omega_{c_0} |\pi_{c_0}(X)| + \sum_{\substack{c_1 = \{1\} \cup c_0 \\ c_0 \subseteq \{2,\ldots,n\}}} \omega_{c_1} |\pi_{c_1}(X)|$$

Since $X = \bigcup_{t \in I} \{t\} \times X(t)$, we have that,

$$\pi_{c_1}(X) = \bigcup_{t\in I} \{t\} \times \pi_{c_0}(X(t)),$$

and hence

$$|\pi_{c_1}(X)| = \int_a^b |\pi_{c_0}(X(t))|\,dt\ .$$

Also

$$\pi_{c_0}(X) = \pi_{c_0}\Big(\bigcup_{t\in I} \{t\} \times X(t)\Big)$$

$$= \pi_{c_0}\Big(\bigcup_{t\in I} X(t)\Big).$$

Therefore

$$\|X\| = \sum_{c_0 \subseteq \{2,\ldots,n\}} \omega_{c_0} \Big|\pi_{c_0}\Big(\bigcup_{t\in I} X(t)\Big)\Big|$$

$$+ \sum_{\substack{c_1 = \{1\} \cup c_0 \\ c_0 \subseteq \{2,\ldots,n\}}} \omega_{c_1} \int_a^b |\pi_{c_0}(X(t))|\,dt$$

$$= \int_a^b \sum_{\substack{c_1 = \{1\} \cup c_0 \\ c_0 \subseteq \{2,\ldots,n\}}} \omega_{c_1} |\pi_{c_0}(X(t))|\,dt$$

$$+ \sum_{c_0 \subseteq \{2,\ldots,n\}} \omega_{c_0} \Big|\pi_{c_0}\Big(\bigcup_{t\in I} X(t)\Big)\Big|$$

$$= \int_a^b \|X(t)\|_1 \, dt + \|\bigcup_{t\in I} X(t)\|_0 .^{\dagger}$$

Letting $f(t) = \int_a^t |X(\tau)|d\tau$, $t \in I$, we see that since $|X(\tau)|$ is non-vanishing for $\tau \in I$, $f(t)$ is monotone in this region. Hence in the expression for $\|X\|$, we can change variables from distance t, to volume $v = f(t)$.

---

† This may in some sense be regarded as the canonical decomposition of the average projection of X. Rivest [14] also starts with a similar expression in his proofs which then proceed along the following lines.

Let $\bar{t}$ be a point such that

$$|X(\bar{t})| \geq |X(t)| \qquad \forall\, t \in I.$$

then

$$\|X\| \geq \int_a^b \|X(t)\|_1 dt + \|X(\bar{t})\|_0,$$

and hence,

$$\min_{\substack{X_n \subseteq [0,1]^n \\ |X_n| = 1/p}} \|X_n\| \geq \min_{\int_a^b h(t)dt = 1/p} \left[\int_a^b \left(\min_{|X_{n-1}(t)| = h(t)} \|X_{n-1}(t)\|_1 \right) dt \right. \tag{9}$$

$$\left. + \min_{|X_{n-1}(\bar{t})| = \max_t h(t)} \|X_{n-1}(\bar{t})\|_0 \right].$$

Now in case all weights ω_c are equal, $\|X_{n-1}(t)\|_1 = \|X_{n-1}(t)\|_0$. Substituting the latter in (9), he obtains a recursion inequality relating the minimum average projection in n dimensions, $\|X_n^*\|$, to the minimum average projection in n-1 dimensions, $\|X_{n-1}^*(t)\|_0$.

Replacing all inequalities by equalities, one can solve the resulting recursion formulae obtaining a lower bound on $\|X_n\|$. This lower bound then turns out to agree with the average projection of a cubical set of size 1/p.

It follows that the above line of reasoning will yield optimal solutions, if and only if, the corresponding recursion inequalities (9) are tight. However, in a general situation in which the weights ω_c are arbitrary, this is not necessarily the case. (The interested reader may utilize Theorem 1 to verify this fact for the example to be given in section III. 6.)

Since $dt = \frac{dv}{|X(v)|}$ we have,

$$\|X\| = \int_0^{1/p} \|X(v)\|_1 \frac{dv}{|X(v)|} + \| \bigcup_{v \in f(I)} X(v) \|_0 . \tag{10}$$

Now by a well known property of averages, there exists a point

$$v^* = f(t^*), \qquad t^* \in I$$
$$v^* \in f(I)$$

with the property that

$$\frac{\|X(v^*)\|_1}{|X(v^*)|} \leq p \int_0^{1/p} \frac{\|X(v)\|_1}{|X(v)|} dv. \tag{11}$$

Combining (10) and (11) above, and using the fact that $\| \bigcup_{v \in f(I)} X(v) \|_0 \geq \|X(v^*)\|_0$, we have

$$\|X\| \geq \frac{1}{p} \frac{\|X(v^*)\|_1}{|X(v^*)|} + \|X(v^*)\|_0$$

$$= \frac{1}{p\,|X(t^*)|} \|X(t^*)\|_1 + \|X(t^*)\|_0$$

$$= \|Y^{n-1}\| \qquad \text{where} \qquad Y^{n-1} = I_1 \times X(t*),$$

and I_1 is any subset of the real line satisfying $|I_1| = \frac{1}{p|X(t*)|}$. Thus there exists a set Y^{n-1} of the form $I_1 \times X(t*)$, with no larger average projection than that of X , but with the same volume $1/p$.

The same proof can be used inductively to show that: if there exists a set,

$$Y^{n-k} = I_1 x \ldots x I_k \text{ x } X(t^*_1,\ldots,t^*_k),\; X(t^*_1,\ldots,t^*_k) \subset R^{n-k}$$

such that $\quad |Y^{n-k}| = |X|$

and $\quad \|Y^{n-k}\| \leq \|X\|$

then there exists a set,

$$Y^{n-k-1} = I_1 x \ldots x I_{k+1} \text{ x } X(t^*_1,\ldots,t^*_{k+1})$$
$$X(t^*_1,\cdots,t^*_{k+1}) \subset R^{n-k-1}$$

such that

$$|Y^{n-k-1}| = |Y^{n-k}| = |X|$$

and

$$\|Y^{n-k-1}\| \leq \|Y^{n-k}\| \leq \|X\|.$$

But then for $k = n-1$, $Y^{n-k} = Y^1$ is rectangular . qed.

Proof of Theorem 1

Theorem 1 now follows as a trivial corollary of Lemma 2.

Corollary 1

If the weights $\omega_{\{1\}},\ldots,\omega_{\{n\}}$ for single attribute queries are positive then,

1. There exists a rectangular optimal solution to the problem (7):

$$\text{minimize} \sum_{c \subseteq \{1,\ldots,n\}} \omega_c |\pi_c(X)|$$

such that $X \subset R^n$, $|X| = 1/p$.

2. The dimensions $s_1, s_2, \ldots, s_n$ of an optimal rectangle are unique and are given by the solution to the local optimality conditions:

$$\frac{\partial}{\partial s_j} \sum_{c \subseteq \{1,\ldots,n\}} \omega_c \prod_{i \in c} s_i = \lambda/ps_j = \mu/s_j \qquad 1 \le j \le n,$$

$$\prod_1^n s_j = 1/p \; . \tag{12}$$

Proof

1. By Lemma 2, there exists an optimal solution to the problem iff there exists a rectangular optimal solution. Thus in searching for an optimal solution we can restrict our attention solely to rectangular sets $I_1 x \cdots x I_n$, $I_i \subset R$, which for our purposes are completely characterized by their dimensions $s_i = |I_i|$. But the average projection of such a rectangular set on hyperplanes defined by a cartesian product of the coordinates is just,

$$\sum_{c \subseteq \{1,\ldots,n\}} \omega_c \prod_{i \in c} s_i . \tag{13}$$

Hence we need to prove that if the weights for single attribute queries are positive, then the problem

$$\text{minimize} \quad \sum_{c \subseteq \{1,\ldots,n\}} \omega_c \prod_{i \in c} s_i$$

$$\text{such that} \; \prod_1^n s_i = 1/p \text{ (and therefore } s_i > 0 \text{ for all } i,\ 1 \le i \le n)$$

has an optimal solution. The latter is a simple problem in geometric programming [6] and can be analyzed more simply if we use the transformations,

$$z_i = \ell g_e \, s_i ,$$

to obtain an equivalent problem:

$$\text{minimize } u(z) = u(z_1,\ldots,z_n) = \sum_{c \subseteq \{1,\ldots,n\}} \omega_c \, e^{\sum_{i\in c} z_i}$$

$$\text{such that } \sum_1^n z_i = -\ell g_e p. \tag{14}$$

Now let $\bar{z}$ be any feasible solution to the above problem, and let $\mu = u(\bar{z})$. Then any other feasible solution $z = (z_1,\ldots,z_n)$, for which $u(z) \leq \mu$, must satisfy:

$$\omega_{\{i\}} \, e^{z_i} \leq \mu \qquad 1 \leq i \leq n.$$

or $\qquad z_i \leq \mu_i, \qquad$ where $\mu_i = \ell g\, \mu - \ell g\, \omega_{\{i\}}$, $1 \leq i \leq n$.

Thus z is bounded in the compact region Z_μ of R^n defined by:

$$\sum_1^n z_i = -\ell g_e p$$

$$z_i \leq \mu_i \qquad 1 \leq i \leq n.$$

Now $Z_\mu \neq \phi$, since $\bar{z} \in Z_\mu$.

Hence, since u is a continuous function, it attains a minimum in Z_μ, which is also the global minimum of u for all $z \in R^n$ such that

$$\sum_1^n z_i = -\ell g_e p.$$

qed.

2. Using the arithmetic-geometric inequality, it can be shown easily that the functions

$$e^{\sum_{i\in c} z_i} \quad \text{are convex.}$$

Hence the objective function in (14) is convex. Furthermore if the singleton weights, $\omega_{\{i\}}$ $1 \le i \le n$, are positive, then it is easy to see that the objective function is in fact strictly convex. Since the feasible region $\{z|\sum_1^n z_i = -\ell g\ p\}$ is also convex, it follows that any locally optimum solution is the unique global optimum to the problem, and hence can be found using the local optimality conditions (12).

qed.

3. <u>Summary</u>

In this section we have shown that the infimum of (6):

$$\hat{a}_p = \inf_{\substack{X \subset R^n \\ |X| = 1/p}} \sum_{c \subseteq \{1,\ldots,n\}} \omega_c |\pi_c(X)|$$

can be found by considering only rectangular sets in R^n.

By Corollary 1, there is a very rich class of parameter values, ω_c ($c \subseteq \{1,\ldots,n\}$), for which this infimum is in fact achievable, and therefore can be found using well known optimization techniques in R^n.

If an optimal rectangle yielding the above infimum fits within the key space $[0,1]^n$, then the lower bound $p\hat{a}_p$ on the average number of pages accessed to answer basic queries can be expected to be approximately tight: one can expect to be able to partition the unit n-cube into approximately optimal rectangles, obtaining a hashing scheme with near minimal retrieval time. Computational results presented in the next section, tend to confirm this belief.

Unfortunately, however, in rare cases where the number of pages is too small, and/or, the query weights are too lopsided, optimal parts of R^n may not fit within the key space $[0,1]^n$. In such cases the results of this paper merely provide a (perhaps weak) lower bound on the average number of pages accessed to answer basic queries.

We conjecture, however, that in such cases the quantity (4),

$$a_p^* = \inf_{\substack{X \subset [0,1]^n \\ |X| = 1/p}} \sum_{c \subseteq \{1,\ldots,n\}} \omega_c |\pi_c(X)|$$

is again attainable by a rectangular subset of $[0,1]^n$, and that the lower bound (5), pa_p^*, will again be approximately tight.

III. Conclusions

1. Summary

A hash function can be characterized by the way it partitions a given attribute-value space K into parts each of which corresponds to a page in memory.

We have considered a model in which K is a rectangular subset of R^n, from which a given number of records are chosen independently of one another and randomly according to a continuous probability density $P_K(\underline{x})$, which is independent in the n attributes. In this context we have looked at the problem of efficiently answering basic queries.

In case pages in memory have equal size, it is natural to restrict one's attention to partitions of the key space K into parts of equal probability; such partitions are known as balanced partitions.

The problem is then to find a balanced partition of the key space K yielding minimal average page access in answering basic queries. In taking this average we have assumed, for each subset c of the attributes, an arbitrary probability ω_c that a query specifies all and only the attributes in c. Also, for queries specifying a given subset of the attributes, the assumption is that the query probability density at a given point, say $v_{i_1},\ldots,v_{i_j}$, is proportional to the probability density that a record takes on the values $v_{i_1},\ldots,v_{i_j}$.

To solve the above problem, given an arbitrary data probability density $P_K(\underline{x}) = P_{K_I}(x_1)\times\cdots\times P_{K_n}(x_n)$, we first show that it is sufficient to consider only the case in which,

$$K = [0,1]^n, \text{ and } P_K(\underline{x}) = 1.$$

Letting p be the number of pages in memory necessary to store the records in a file, it is then easy to see that the quantity, p x (minimum average projection of a part of R^n of size 1/p, on a cartesian product of coordinates) is a lower bound on the minimum average page access for the latter problem.

In section 2 we showed that, in order to find the minimum average projection of a part of R^n, of a fixed size, it is sufficient to consider only rectangular parts of R^n. It follows that if the key space $[0,1]^n$, can be partitioned into such optimal rectangular parts, then the induced hash function achieves minimum average page access.
This, however, may not be possible: in the first place, an optimal part of R^n may not fit within our key space $[0,1]^n$ (if a side of an optimal rectangle happens to be greater than 1); secondly, even if such parts do fit within the unit n-cube, it may not be possible to cover the latter with disjoint optimal parts.

The former problem is serious and requires further research, but seems to occur rather infrequently for large p.

The latter is somewhat less serious. If an optimal part fits within the unit n-cube, then it is not hard to convince oneself that it will be possible in a majority of cases, to partition the unit n-cube into approximately optimal parts, given that the parts are reasonably small (that the number of pages is reasonably large.)

2. Computational Results.

To test the seriousness of the above problems, we computed the dimensions of an optimal part of R^n, for a number of cases in 3 and 4 dimensions. Twenty trials were made in each case, for which the query weights (unnormalized probabilities) were chosen randomly once from the integers 1,2,...,100, and again from the integers 1, 10, and 100. The results of these computations are summarized in Table 1.

As can be seen, the number of solutions that are out of bounds i.e. for which at least one side of an optimal rectangle is greater than 1, is quite small. In cases where optimal parts are inbounds, it is of course extremely unlikely that the key space $[0,1]^n$ can be covered by disjoint optimal rectangles, i.e. that each side of an optimal rectangle is an integral divisor of 1. We therefore tried to find slightly smaller rectangular parts of approximately the same shape, whose sides were in fact integral divisors of 1. This 'roundoff' process introduces two sources of error. On the one hand since the parts are smaller than 1/p, the number of parts is greater than p and hence more pages are required in memory: an extra storage cost. On the other hand, since there are more pages in total, the average page access may be larger than the minimum possible using p pages: an extra time cost. The latter error is estimated by comparing the average page access after roundoff, with the derived lower bound (5) for p pages.

These roundoff errors, however, were again found to be small: less than 10% in all cases considered and less than 5% in most.

DIMENSION (n)	NUMBER OF PAGES (p)	QUERY WEIGHT SELECTION	NUMBER OF TRIALS	ROUNDOFF ERRORS STORAGE		ROUNDOFF ERRORS TIME		NUMBER OF SOLUTIONS OUT OF BOUNDS
				≤ 5%	5% < ≤ 10%	≤ 5%	5% < ≤ 10%	
3	1,000	Random From 1,2,3,...,100	20	16	4	20	0	0
3	1,000	Random From 1,10,100	20	18	0	18	0	2
4	1,000	Random From 1,2,3,...,100	20	9	11	12	8	0
4	10,000	Random From 1,2,3,...,100	20	14	6	19	1	0
4	1,000	Random From 1,10,100	20	11	6	15	2	3
4	10,000	Random From 1,10,100	20	15	2	17	0	3

Table 1

3. Near Optimal Hash Functions for a Relation with Independent Attributes

Using Lemma 1 (section 1.4), we can now characterize near optimal partitions of a general relation with independent attributes.

In particular what needs to be done is to invert the function g: $K \to [0,1]^n$ used in proving the lemma, and to consider the corresponding induced mapping from partitions of $[0,1]^n$ onto partitions of K. Again, the details are trivial and will be left to the reader.

We are then in a position to conclude that for the model presented in this paper, the problem of finding optimum hash functions, is adequately solved in many cases as follows.

Step 1

Use the query probabilities ω_c and the number of pages p to solve the equations (12) or otherwise find optimal parts (rectangles) of R^n of size 1/p.

Step 2

In many situations the dimensions of an optimal part of R^n will be less than 1 for reasonably large p. If this is the case, then find a rectangular part of R^n, of slightly smaller size, but approximately the same shape, whose sides are integral divisors of 1, say

$$\frac{1}{N_1}, \frac{1}{N_2}, \ldots, \frac{1}{N_n} .$$

Step 3

In the original key space K divide each coordinate i, into N_i parts of equal probability, according to the probability density $P_{K_i}(x_i)$. Let these parts be denoted by,

$$I^i_{j_i}, \quad 1 \leq j_i \leq N_i, \quad 1 \leq i \leq n.$$

The cartesian products of such parts in different coordinates,

$$I^1_{j_1} \times I^2_{j_2} \times \cdots \times I^n_{j_n}$$

$$1 \leq j_1 \leq N_1, \ldots\ldots, \quad 1 \leq j_n \leq N_n,$$

form a balanced partition of the attribute-value space into a number of parts that is slightly greater than p.† In most situations the extra number of pages needed for storage, and the extra number of pages accessed per query in the induced hashing scheme is small, particularly if one takes into consideration the fact that it is desirable to leave some slack in each page to avoid excessive overflow. Hence in most cases the induced hashing scheme has near minimal average page access.

4. Problems with Discrete Data

Throughout this paper it has been assumed that the given attribute-value space is intrinsically continuous in nature. In case of discrete data our analysis will work for any continuous approximation of the discrete data and query probabilities, and hence the resulting lower bounds will still be valid. However, if the number of possible values of an attribute is too small, then it may not be possible even to approximate step 3 of the above procedure. Thus, in cases where a small number of values of an attribute occur most frequently, a more detailed analysis of the situation is required. (See Rivest [14] for the uniform binary case.)

---

† Hash functions induced by such partitions were first introduced by Rothnie and Lozano [15], where they are referred to as 'multiple key hash functions.'

5. Ease of Addressing

In considering the problem of finding optimal hash functions we have so far chosen to ignore problems of addressing, or the process of computing a desired hash function on the one hand, and of finding the storage locations of pages that need to be accessed to answer a query, on the other hand. This process, of course, may in itself require the storage and retrieval of parameter values used in address calculation, which will have to be accounted for in any overall evaluation of the efficiency of a hashing scheme.

We claim, however, that the simple rectangular hash functions derived above, lend themselves to easy addressing, with relatively small expenditures of storage and time, as compared with memory and time requirements for the storage and retrieval of the actual data.

For a justification of this claim we refer the reader to Rothnie and Lozano [15]. Briefly stated, a rectangular (multiple key) hash function on an n-dimensional attribute-value space can be thought of as a composition of n simple hash functions, one for each attribute. The mechanism for computing any one of these hash functions would then be identical to that used in address calculation for retrieval via a single (primary) key, about which a great deal is known. One may for example use a simple, possibly randomizing, function of an attribute (see Knuth [11] Section 6.4, Hashing), or an order-preserving function of an attribute, whose computation can be aided by a tree structure (see Knuth [11] Section 6.2.4 Multiway Trees, and Held and Stonebraker [8] Section 9, Generalized Directories).

6. Example

We illustrate the process of finding near optimal hash functions as described above, by the following example in 3 dimensions. Let us consider the simplified DMV file in which there is a record for each registered automobile in California, and each record has the form,

$$(x_1, x_2, x_3)$$

where

$x_1 \in F = \{\text{family names}\}$

$x_2 \in C = \{\text{cities in California}\}$

$x_3 \in M = \{\text{auto manufacturers}\}$, and x_1, x_2, and x_3, occur independently of one another.

A record may, of course, contain other pertinent information about a registered automobile. We shall assume however, that such information is not used for record selection, so that its exclusion does not alter the analysis.

There are 8 basic query types associated with this file, corresponding to the 8 subsets of the set of attributes {1,2,3}. We represent these query types mnemonically by,

θ , 1, 2, 3, 12, 13, 23, and 123.

(The query type θ contains a single query asking for all the records in the file, while a query of type 12 say, may be, retrieve all records for which,

last name = Smith   and   city of residence = Berkeley.)

Suppose by monitoring incoming queries for a period of time, we determine the following query probabilities:

$\omega_\theta = 0$, $\omega_1 = 100/142$ $\omega_2 = 1/142$ $\omega_3 = 10/142$

$\omega_{12} = 10/142$, $\omega_{13} = 1/142$, $\omega_{23} = 10/142$, $\omega_{123} = 10/142$.

Suppose further that the volume of data at hand requires 1000 pages for its storage. In that case,

1. From (13) to find optimal parts of R^3 we need to minimize,

$$100\ s_1 + s_2 + 10\ s_3 + 10\ s_1s_2 + s_1s_3 + 10\ s_2s_3 + 10\ s_1s_2s_3 \tag{16}$$

such that $\quad s_1s_2s_3 = 1/1000.$

The last term $10\ s_1s_2s_3$ is a constant (= 0.01) and does not enter into the minimization. Hence corresponding to the local optimality conditions (12) we have:

$$100 + 10\ s_2^* + s_3^* = \mu/s_1^*$$

$$1 + 10\ s_1^* + 10\ s_3^* = \mu/s_2^*$$

$$10 + s_1^* + 10\ s_2^* = \mu/s_3^*$$

$$s_1^*s_2^*s_3^* = 1/1000$$

Solving these, we find the dimensions of an optimal part of R^3 to be:

$$s_1^* \approx 0.0146$$

$$s_2^* \approx 0.775$$

$$s_3^* \approx 0.0887$$

Notice that such an optimal rectangle in R^3 can fit within the unit n-cube. This means that the minimum average number of times a page, corresponding to a part of the key space, FxCxM, of probability 1/1000, is accessed to answer a query is,

$$1/142\ (100\ s_1^* + s_2^* + 10\ s_3^* + 10 s_1^*\ s_2^* + s_1^* s_3^* + 10 s_2^* s_3^* + 0.01) = 0.0277.$$

Hence the lower bound (5) implies that the average page access is bounded below by

$$1000 \times 0.0277 = 27.7 \text{ pages/query.}$$

2. As expected s_1^*, s_2^*, s_3^* are not integral divisors of 1 and hence the unit cube cannot be partitioned into 1000 optimal pages. However taking,

$$s_1 = \frac{1}{91} = 0.011$$

$$s_2 = \frac{1}{1} = 1.0$$

$$s_3 = \frac{1}{11} = 0.091$$

we obtain slightly smaller parts of size 0.000999, but of approximately the same shape, 1001 of which will exactly partition the unit cube. This roundoff process requires 1 extra page, or 0.1% more storage space. Also the average page access for this rectangular partition is

$$1001(100\ s_1 + s_2 + 10\ s_3 + 10\ s_1 s_2 + s_1 s_3 + 10\ s_2 s_3 + 0.01)/142$$

= 28.5 pages/query, or 2.9% more than the derived lower bound, 27.7 pages/query.

3. Thus in order to find an approximately optimal partition of the original attribute-value space, FxCxM, we first divide the last name field into 91 intervals of equal probability, F_1, F_2,..., F_{91}. (Each interval F_k, will contain the last names of owners of 1/91, or 1.1%, of automobiles in California.) We then divide the make of car field into 11 intervals M_1,..., M_{11}, of equal probability.

The rectangles

$$F_k x C x M_\ell \qquad 1 \leq k \leq 91$$
$$1 \leq \ell \leq 11$$

then form a near optimal partition of the key space FxCxM, and the corresponding hash function achieves approximately minimal average retreival time.

Acknowledgement

I would like to express my thanks to Professor R.M. Karp for his careful attention and criticism during many hours of discussion leading up to the present work, and to Professor L.A. Zadeh for his patient guidance, support, and encouragement.

I would also like to thank Colin McMaster for reading parts of the final manuscript, and Carlos Humes for introducing me to Geometric Programming.

The earlier work of Ronald L. Rivest on "The Analysis of Associative Retrieval Algorithms" provides a basis for the present work. Reading it has been an enjoyable and an instructive experience, and its influence is evident throughout this paper.

References

1. Abraham, C.T., S.P. Ghosh, and D.K. Ray-Chaudhuri, File Organization Schemes Based on Finite Geometries. Information and Control 12 (February 1968), 143-163.

2. Blaschke, W., Kreis und Kugel (Circle and Sphere). Chelsea Publishing Co. N.Y. 1949.

3. Bose, R.C. and Gary G. Koch, The Design of Combinatorial Information Retrieval Systems for Files with Multiple Valued Attributes. SIAM Journal of Applied Mathematics 17 (Nov. 1969), 1203-1214.

4. Chow, D.K., New Balanced File Organization Schemes. Information and Control 15(1969), 377-396.

5. Codd, E.F., A Relational Model of Data for Large Shared Data Banks. CACM 13.6 (June 1970) 377-387.

6. Duffin, R.J., E.L. Peterson, and Clarence Zener, Geometric Programming. John Wiley & Sons, Inc. N.Y.

7. Gustafson, R.A., Elements of the Randomized Combinatorial File Structure. Proceedings of the Symposium on Information Storage and Retrieval, ACM SIGIR, University of Maryland (April 1971), 163-174.

8. Held, G., and Michael Stonebraker, Storage Structures and Access Methods in the Relational Data Base Management System INGRES. Proc. of the Pacific 75 Regional Conf. ACM (April 1975), 26-33.

9. Hsiao, David and Frank Harary, A Formal System of Information Retrieval from Files. CACM 13 (Feb. 1970), 67-73.

10. Kazarinoff, N.D., Geometric Inequalities. Random House New Mathematical Library #4, 1961.

11. Knuth, Donald E., The Art of Computer Programming, Volume 3, Sorting and Searching. Addison Wesley (1972).

12. Lefkovitz, David, File Structures for On-Line Systems. Spartan Books (1969).

13. Lum, Vincent Y., Multi Attribute Retrieval with Combined Indices. CACM 13.11 (Nov. 1970).

14. Rivest, Ronald L., Analysis of Associative Retrieval Algorithms. Rapport de Recherche no. 54 (Feb. 1974), Laboratoire de Recherche en Informatique et Automatique, Rocquencourt, France.

15. Rothnie, J.B., Jr. and T. Lozano, Attribute Based File Organization in a Paged Memory Environment. CACM 17.2 (Feb. 1974), 63-69.

16. Welch, Terry, Bounds on Information Retrieval Efficiency in Static File Structures. Technical Report MAC TR-88, Project MAC MIT, Cambridge, Mass. 02139.

17. Wong, E. and T.C. Chiang, Canonical Structure in Attribute Based File Organization. CACM 14 (Sept. 1971), 593-597.