Performance Analyses of Cartesian Product Files and Random Files

C. C. CHANG, M. W. DU, MEMBER, IEEE, AND R. C. T. LEE

Abstract—**In this paper, we shall derive two formulas for the average number of buckets to be examined over all possible partial match queries for Cartesian product files and random files, respectively. The superiority of the Cartesian product file is established. A new multikey file, called a partition file, is introduced. It is shown that both Cartesian product files and random files are special cases of partition files.**

Index Terms—**Cartesian product files, partial match queries, partition files, random files.**

I. Introduction

IN THIS PAPER, we are concerned with some problem of designing optimal multiattribute file systems for partial match queries [1]-[4], [6]-[11]. By a multiattribute file system, we mean a file system whose records are characterized by more than one attribute. By partial match queries, we mean queries of the following form: retrieve all records where $A_{i_1} = a_{i_1}, A_{i_2} = a_{i_2}, \cdots, A_{ij} = a_{i_j}$ and $i_1 \neq i_2 \neq \cdots \neq i_j$.

In this paper, we shall limit ourselves to the case where all possible records are present. Note that every record is characterized by N-attributes $A_1, A_2, A_3, \cdots, A_N$. Let the domain of attribute A_i be denoted as D_i. Thus the set of all possible records is $D_1 \times D_2 \times \cdots \times D_N$. In the rest of this paper, whenever we discuss the partial match problem, we shall assume that every possible record in this set $D_1 \times D_2 \times \cdots \times D_N$ is present.

We shall assume that every file is divided into buckets. The problem of multiattribute file design can be explained by considering the two file systems shown in Tables I and II, respectively.

In both tables, a query $(a, *)$ denotes a query retrieving records with the first attribute equal to a and the second attribute with any value. Similarly, for a 3-attribute file system, a query denoted as $(*, b, c)$ denotes a query retrieving all records with $A_2 = b$ and $A_3 = c$ and A_1 can be of any value. The reader can see that the average number of buckets to be examined, over all possible queries, is two for the file in Table II and four for that in Table I.

Thus the problem of multiattribute file system design for partial match queries is as follows: given a set of multiattribute records, arrange the records into the NB buckets in such a way that the average number of buckets to be examined, over all possible partial match queries, is minimized.

In [12], it was pointed out that the multiattribute file system design problem as stated above is an NP-complete problem [5]. Thus it must be considered as a very difficult problem. However, as shown in [8], in certain restricted case, we may still have good file systems for partial match queries.

TABLE I
A BAD FILE STRUCTURE

Bucket 1	Bucket 2	Bucket 3	Bucket 4
(a,a)	(a,b)	(a,c)	(a,d)
(b,b)	(b,c)	(b,d)	(b,a)
(c,c)	(c,d)	(c,a)	(c,b)
(d,d)	(d,a)	(d,b)	(d,c)

(a)

Queries	Buckets to be examined
(a,*)	1, 2, 3, 4
(b,*)	1, 2, 3, 4
(c,*)	1, 2, 3, 4
(d,*)	1, 2, 3, 4
(*,a)	1, 2, 3, 4
(*,b)	1, 2, 3, 4
(*,c)	1, 2, 3, 4
(*,d)	1, 2, 3, 4

(b)

TABLE II
A GOOD FILE STRUCTURE

Bucket 1	Bucket 2	Bucket 3	Bucket 4
(a,a)	(a,c)	(c,a)	(c,c)
(a,b)	(a,d)	(c,b)	(c,d)
(b,a)	(b,c)	(d,a)	(d,c)
(b,b)	(b,d)	(d,b)	(d,d)

(a)

Queries	Buckets to be examined
(a,*)	1, 2
(b,*)	1, 2
(c,*)	3, 4
(d,*)	3, 4
(*,a)	1, 3
(*,b)	1, 3
(*,c)	2, 4
(*,d)	2, 4

(b)

II. Cartesian Product Files and Random Files

Multiattribute file system design for partial match queries has been considered by many authors. Rivest [10] suggested

Manuscript received October 30, 1981; revised March 8, 1983. This work was supported in part by the National Science Council, Republic of China, under Contract NSC-69E-0404-03(06).

C. C. Chang and M. W. Du are with the Institute of Computer Engineering, National Chiao-Tung University, Hsinchu, Taiwan, Republic of China.

R. C. T. Lee is with the Institute of Computer and Decision Sciences, National Tsing Hua University, Hsinchu, Taiwan, Republic of China.

the string homomorphism hashing (SHH for short) method. Rothnie and Lozano [11] suggested the multikey hashing (MKH for short) method. Liou and Yao [9] suggested the multidimensional directory (MDD for short) method. Lee and Tseng [7] suggested the multikey sorting (MKS for short) method. Aho and Ullman [1] explored the problem of designing optimal multiattribute file systems whose probabilities of an attribute being specified are not equal.

In [8], it was proved that all of those file designing methods exhibit one common property: records in one bucket are similar to one another. In [8], it was also pointed out that, under certain conditions, the file systems designed by using the SHH, MKH, and MDD methods are all Cartesian product files. The file system explored by Aho and Ullman [1] is also a Cartesian product file. Moreover, [3] proposed a new hash scheme which can be used to produce any arbitrary Cartesian product file. The Cartesian product files are defined as follows.

Definition: Let there be N attributes $A_1, A_2, \cdots, A_N$. Let the domain of A_i be D_i. Let each domain D_i be divided into m_i subdomains $D_{i1}, D_{i2}, \cdots, D_{im_i}$. A Cartesian product file is a file in which the records in each bucket are of the form $D_{1S_1} \times D_{2S_2} \times \cdots \times D_{NS_N}$.

Example 2.1: Let $D_1 = \{a, b, c, d\} = D_2$. Let $D_{11} = \{a, b\} = D_{21}$. Let $D_{12} = \{c, d\} = D_{22}$. Then the following file is a Cartesian product file:

Bucket 1: $D_{11} \times D_{21} = \{(a,a), (a,b), (b,a), (b,b)\}$

Bucket 2: $D_{11} \times D_{22} = \{(a,c), (a,d), (b,c), (b,d)\}$

Bucket 3: $D_{12} \times D_{21} = \{(c,a), (c,b), (d,a), (d,b)\}$

Bucket 4: $D_{12} \times D_{22} = \{(c,c), (c,d), (d,c), (d,d)\}$.

The reader can see that the above file system is exactly the same file system shown in Table II.

Example 2.2: Let $D_1 = \{a, b, c, d, e\}$ and $D_2 = \{a, b, c, d\}$. Let $D_{11} = \{a, b, c\}$, $D_{12} = \{d, e\}$, $D_{21} = \{a, b\}$, and $D_{22} = \{c, d\}$. Then the following file system is a Cartesian product file system:

Bucket 1: $D_{11} \times D_{21} = \{(a,a), (a,b), (b,a), (b,b), (c,a), (c,b)\}$

Bucket 2: $D_{11} \times D_{22} = \{(a,c), (a,d), (b,c), (b,d), (c,c), (c,d)\}$

Bucket 3: $D_{12} \times D_{21} = \{(d,a), (d,b), (e,a), (e,b)\}$

Bucket 4: $D_{12} \times D_{22} = \{(d,c), (d,d), (e,c), (e,d)\}$.

Note that in this case, the number of records in Bucket 1 is not the same as that in Bucket 3.

The concept of Cartesian product file can also be explained from the geometry point of view. The file system of Example 2.2 is now depicted in Fig. 1. We may say that the records in each bucket are highly correlated to one another.

Consider Fig. 2. In this case, we have randomly selected six records and put them into Bucket 1. Buckets 2, 3, and 4 will be constructed likewise. In other words, records in each bucket are selected totally randomly. These kinds of files are called random files.

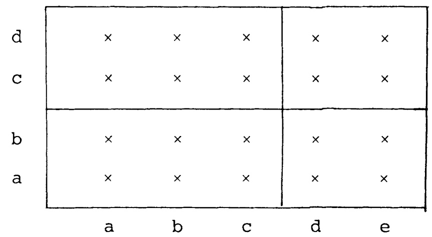


Fig. 1. A Cartesian product file.

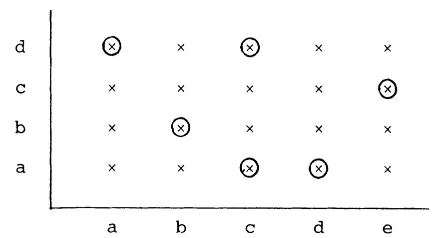


Fig. 2. A random file.

In this paper, we shall investigate the performance of Cartesian product files and the performance of random files. We shall derive formulas relating to the performances of these files. We shall then show that Cartesian product files always perform better than random files. Finally, a general model for multikey files is shown. Both Cartesian product files and random files can be seen as special cases of this general model.

III. The Performance of Cartesian Product Files and Random Files

Let us assume that our records are characterized by N attributes $A_1, A_2, \cdots, A_N$ and the domain of A_i is D_i. Let the number of elements in D_i be denoted as d_i. Then the number of records NR is equal to $d_1 d_2 \cdots d_N$. Let NB denote the number of buckets. Then the bucket size BZ is equal to NR/NB. We shall assume that BZ is an integer. Let ANB_{CD} and ANB_R denote the expected number of buckets being accessed over all possible partial match queries in a Cartesian product file and a random file, respectively.

If a file is a Cartesian product file, for every bucket, records in this bucket are of the form of $D_{1S_1} \times D_{2S_2} \times \cdots \times D_{NS_N}$ where D_{jS_j} is a subset of D_j. Let the domain size of D_{jS_j} be denoted as z_j. To simplify our discussion, we shall assume that z_j is the same for every bucket. Note that this is not the case for the file shown in Example 2.2. In this case, $z_1 = 3$ for Bucket 1 and $z_1 = 2$ for Bucket 3. It is much too complicated for our discussion.

For a Cartesian product file, we now ask: what is the number of queries which need to examine a bucket in the file? (Note that this answer is true for every bucket in a Cartesian product file.) The answer is as follows.

1) There are $z_1 + z_2 + \cdots + z_N$ partial match queries which involve exactly one attribute.

2) There are $z_1 z_2 + z_1 z_3 + \cdots + z_{N-1} z_N$ partial match queries which involve exactly two attributes.

3) There are $z_1 z_2 \cdots z_{N-1} + \cdots + z_2 z_3 \cdots z_N$ partial match queries which involve exactly $N-1$ attributes.

Totally, for each bucket in a Cartesian product file, the total number of partial match queries which need to examine this bucket is

$$\begin{aligned} & z_1 + z_2 + \cdots + z_N \\ & + z_1 z_2 + z_1 z_3 + \cdots + z_{N-1} z_N \\ & + \cdots \\ & + z_1 z_2 \cdots z_{N-1} + \cdots + z_2 z_3 \cdots z_N . \end{aligned}$$

In other words, the number of queries which need to examine this bucket is

$$\sum_{j=1}^{N-1} \sum_{\substack{\{z_{i_1}, z_{i_2}, \cdots z_{i_j}\} \in \{z_1, z_2, \cdots z_N\} \\ i_2 < i_2 < \cdots < i_j}} z_{i_1} z_{i_2} \cdots z_{i_j}$$

Hence, for all possible queries, the total number of buckets to be examined is

$$NB \cdot \left(\sum_{j=1}^{N-1} \sum_{\substack{\{z_{i_1}, z_{i_2}, \cdots z_{i_j}\} \in \{z_1, z_2, \cdots, z_N\} \\ i_1 < i_2 < \cdots < i_j .}} z_{i_1} z_{i_2} \cdots z_{i_j} \right)$$

Therefore,

$$ANB_{cp} = \frac{NB}{NQ} \left(\sum_{j=1}^{N-1} \sum_{\substack{\{z_{i_1}, z_{i_2}, \cdots z_{i_j}\} \in \{z_1, z_2, \cdots, z_N\} \\ i_1 < i_2 < \cdots < i_j}} z_{i_1} z_{i_2} \cdots z_{i_j} \right) \quad (3.1)$$

where NQ is the total number of partial match queries.

The total number of partial match queries can be found as follows.

1) There are $d_1 + d_2 + \cdots + d_N$ partial match queries which involve exactly one attribute.

2) There are $d_1 d_2 + d_1 d_3 + \cdots + d_{N-1} d_N$ partial match queries which involve exactly two attributes.

3) There are $d_1 d_2 \cdots d_{N-1} + \cdots + d_2 d_3 \cdots d_N$ partial match queries which involve exactly $N - 1$ attributes.

Let $\{d_{i_1}, d_{i_2}, \cdots, d_{i_j}\}$ be a subset with j elements chosen from $\{d_1, d_2, \cdots, d_N\}$. In general, there are

$$\sum_{\substack{\{d_{i_1}, d_{i_2}, \cdots, d_{i_j}\} \in \{d_1, d_2, \cdots, d_N\} \\ i_1 < i_2 < \cdots < i_j}} d_{i_1} d_{i_2} \cdots d_{i_j}$$

partial match queries which involve exactly j attributes. The total number of queries

$$\begin{aligned} & = d_1 + d_2 + \cdots + d_N \\ & \quad + d_1 d_2 + d_1 d_3 + \cdots + d_{N-1} d_N \\ & \quad + \cdots \\ & \quad + d_1 d_2 \cdots d_{N-1} + \cdots + d_2 d_3 \cdots d_N \\ & = \sum_{j=1}^{N-1} \sum_{\substack{\{d_{i_1}, d_{i_2}, \cdots, d_{i_j}\} \in \{d_1, d_2, \cdots, d_N\} \\ i_1 < i_2 < \cdots < i_j}} d_{i_1} d_{i_2} \cdots d_{i_j} \\ & = NQ. \end{aligned}$$

Example 3.1: Consider the case where $d_1 = 4$, $d_2 = 2$, $d_3 = 3$, $z_1 = 2$, $z_2 = 1$, and $z_3 = 3$.

In this case, the number of buckets (NB) is $(d_1 d_2 d_3)/(z_1 z_2 z_3) = (4 \times 2 \times 3)/(2 \times 1 \times 3) = 4$ and the number of queries (NQ) is $d_1 + d_2 + d_3 + d_1 d_2 + d_1 d_3 + d_2 d_3$. Therefore,

$$\begin{aligned} ANB_{cp} & = \frac{NB}{NQ} (z_1 + z_2 + z_3 + z_1 z_2 + z_1 z_3 + z_2 z_3) \\ & = \frac{4}{4 + 2 + 3 + 4 \times 2 + 4 \times 3 + 2 \times 3} \\ & \quad \cdot (2 + 1 + 3 + 2 \times 1 + 2 \times 3 + 1 \times 3) \\ & = \frac{4}{35} \times 17 = 1.943. \end{aligned}$$

In the following, we shall derive the formula for ANB_R.

Let us consider a special partial match query $A_i = a_i$, where $a_i \in D_i$. There are $d_1 d_2 \cdots d_{i-1} d_{i+1} \cdots d_N$ records satisfying the condition $A_i = a_i$. Since each record is randomly assigned to a bucket, the probability that a bucket contains no record to be searched for this query is $C(NR - d_1 d_2 \cdots d_{i-1} d_{i+1} \cdots d_N, BZ)/C(NR, BZ)$, where $C(M, N)$ denotes the number of N combinations out of M objects. In other words, the probability that a bucket needs to be examined by this particular partial match query is

$$1 - C(NR - d_1 d_2 \cdots d_{i-1} d_{i+1} \cdots d_N, BZ)/C(NR, BZ).$$

The expected number of buckets which need to be examined by this partial match query is $NB(1 - C(NR - d_1 d_2 \cdots d_{i-1} d_{i+1} \cdots d_N, BZ)/C(NR, BZ))$. Note that for all $a_i \in D_i$, all partial match queries $A_i = a_i$ produce the same result:

$$ANB_R = \left(\sum_{\substack{\text{all partial} \\ \text{match queries}}} \text{the expected number of buckets to be accessed for a partial match query/the total number of different partial match queries} \right).$$

Hence

$$ANB_R = \left(\sum_{\substack{\text{all partial} \\ \text{match queries}}} \text{the expected number of buckets to be accessed for a partial match query}/NQ \right).$$

Let TNB_j be the total number of buckets to be accessed over all partial match queries with j attributes being specified.

$$1)\ TNB_1 = \sum_{d_i \in \{d_1, d_2, \cdots, d_N\}} d_i \cdot NB \cdot (1 - C(NR - d_1 d_2 \cdots d_{i-1} d_{i+1} \cdots d_N, BZ)/C(NR, BZ))$$

$$2)\ TNB_2 = \sum_{\substack{\{d_i,d_j\}\in\{d_1,d_2,\cdots,d_N\}\\ i<j}} d_i d_j \cdot NB(1 - C(NR - d_1 d_2 \cdots d_{i-1} d_{i+1} \cdots d_{j-1} d_{j+1} \cdots d_N, BZ)/C(NR, BZ))$$

$$3)\ TNB_{N-1} = d_2 d_3 \cdots d_N \cdot NB(1 - C(NR - d_1, BZ)/C(NR, BZ)) + d_1 d_3 \cdots d_N \cdot NB(1 - C(NR - d_2, BZ)/C(NR, BZ)) + \cdots + d_1 d_2 \cdots d_{N-1} \cdot NB(1 - C(NR - d_N, BZ)/C(NR, BZ)).$$

In general,

$$TNB_j = \Sigma d_{i_1} d_{i_2} \cdots d_{i_j} \cdot NB \cdot (1 - C(NR - NR/d_{i_1} d_{i_2} \cdots d_{i_j}, BZ)/C(NR, BZ))\{d_{i_1}, d_{i_2}, \cdots d_{i_j}\} \in \{d_1, d_2, \cdots, d_N\} i_1 < i_2 < \cdots < i_j$$

and

$$ANB_R = \sum_{j=1}^{N-1} TNB_j/NQ = (NB/NQ)\left(\sum_{j=1}^{N-1} \sum_{\substack{\{d_{i_1},d_{i_2},\cdots,d_{i_j}\}\in\{d_1,d_2,\cdots,d_N\}\\ i_1<i_2<\cdots<i_j}} d_{i_1} \cdot d_{i_2} \cdots d_{i_j}(1 - C(NR - NR/d_{i_1} d_{i_2} \cdots d_{i_j}, BZ)/C(NR, BZ))\right) \quad (3.2)$$

Also,

$$ANB_R = (NB/NQ)\left(NQ - \left(\sum_{j=1}^{N-1} \Sigma d_{i_1} d_{i_2} \cdots d_{i_j} \{d_{i_1}, d_{i_2}, \cdots, d_{i_j}\} \in \{d_1, d_2, \cdots, d_N\}\ i_1 < i_2 < \cdots < i_j \cdot C(NR - NR/d_{i_1} d_{i_2} \cdots d_{i_j}, BZ)/C(NR, BZ)\right)\right)$$

$$= (NB/NQ)\left(NQ - \left(\sum_{j=1}^{N-1} \Sigma(NR/d_{i_1} d_{i_2} \cdots d_{i_j}) \{d_{i_1}, d_{i_2}, \cdots, d_{i_j}\} \in \{d_1, d_2, \cdots, d_N\}\ i_1 < i_2 < \cdots < i_j \cdot C(NR - d_{i_1} d_{i_2} \cdots d_{i_j}, BZ)/C(NR, BZ)\right)\right).$$

Therefore,

$$ANB_R = (NB/NQ)\left(NQ - (NR/C(NR, BZ)) \cdot \left(\sum_{j=1}^{N-1} \Sigma\ 1/d_{i_1} d_{i_2} \cdots d_{i_j} \cdot C(NR - d_{i_1} d_{i_2} \cdots d_{i_j}, BZ)\right)\right).$$

Hence, given d_1, d_2, $\cdots$, d_N and NB, we can calculate ANB_R immediately, where $NR = d_1 d_2 \cdots d_N$ and $BZ = NR/NB$.

Example 3.2: Consider the same case of Example 3.1, where $d_1 = 4$, $d_2 = 2$, $d_3 = 3$, and $NB = 4$. We have $NR = d_1 d_2 d_3 = 4 \times 2 \times 3 = 24$ and $BZ = NR/NB = 24/4 = 6$. $NQ = d_1 + d_2 + d_3 + d_1 d_2 + d_1 d_3 + d_2 d_3 = 35$.

$$\begin{aligned} ANB_R &= (NB/NQ) \cdot (NQ - (NR/C(NR, BZ)) \cdot ((1/d_1) \cdot C(NR - d_1, BZ) + (1/d_2) \cdot C(NR - d_2, BZ) \\ &\quad + (1/d_3) \cdot C(NR - d_3, BZ) + (1/d_2 d_3) \cdot C(NR - d_2 d_3, BZ) + (1/d_1 d_3) \\ &\quad \cdot C(NR - d_1 d_3, BZ) + (1/d_1 d_2) \cdot C(NR - d_1 d_2, BZ))) \\ &= (4/35)(35 - (24/C(24, 6)) \cdot ((1/4) \cdot C(24 - 4, 6) + (1/2) \cdot C(24 - 2, 6) \\ &\quad + (1/3) \cdot C(24 - 3, 6) + (1/(2 \times 3)) \cdot C(24 - 6, 6) \\ &\quad + (1/(4 \times 3)) \cdot C(24 - 12, 6) + (1/(4 \times 2)) \cdot C(24 - 8, 6))) \\ &= (4/35) \times (35 - 3.087) = (4/35) \times 31.913 \\ &= 127.652/35 = 3.6472. \end{aligned}$$

Compare the results of Examples 3.1 and 3.2. We have $ANB_{CP} < ANB_R$ in the same case.

Now we are interested in whether the value of ANB_{CP} is always less than that of ANB_R. The following theorem shows that it is indeed the case.

Theorem 3.1: Let there be N attributes where the domain of each attribute is d_i. Let the number of buckets be NB. Let the Cartesian product file partition each D_i into m_i subdomains. That is, $d_i = m_i z_i$. Then $ANB_{CP} < ANB_R$ if $m_i \geqslant 2$ and $z_i \geqslant 2$ for all $i = 1, 2, \cdots, N$. (The proof of this theorem is quite long and can be found in the Appendix.)

The condition $m_i \geqslant 2$ and $z_i \geqslant 2$ for all $i = 1, 2, \cdots, N$ is sufficient, but not necessary. Consider the following case:

$$
\begin{aligned}
d_1 &= 2,\\
d_2 &= 4,\\
d_3 &= 4,\\
NB &= 4,\\
BZ &= d_1 d_2 d_3 / NB = (2 \cdot 4 \cdot 4)/4 = 8.\\
NQ &= d_1 + d_2 + d_3 + d_1 d_2 + d_1 d_3 + d_2 d_3\\
&= 2 + 4 + 4 + 8 + 8 + 16\\
&= 42.
\end{aligned}
$$

Let $z_1 = 2$, $z_2 = 2$, and $z_3 = 2$. This means that $m_1 = 1$, $m_2 = 2$ and $m_3 = 2$. In this case, ANB_{CP} and ANB_R can be found to be 1.71 and 2.70, respectively. Thus, $ANB_{CP} < ANB_R$. But the above condition is not satisfied.

Note that the cases of $m_i = 1$ or $z_i = 1$ are rather unusual cases. For instance, if $z_i = 1$ for all i, the Cartesian product file reduces to a random file. We are working on a new proof which we hope will cover the cases of $m_i = 1$ or $z_i = 1$.

IV. The Partition File—A General Model for Both Cartesian Product Files and Random Files

In the previous section, we introduced the concept of Cartesian product files and that of random files. These two concepts seem to be entirely different ones. In fact, the formulas concerning with these two concepts bear no resemblance to each other at all.

In this section, we shall introduce a new file structure called the partition file. This file is interesting because it can be shown that Cartesian product files and random files are special cases of this partition file.

In the partition file, we shall assume that the entire set of $d_1 d_2 \cdots d_N$ records are divided into partitions. The partitions are constructed as follows: let $d_i = m_i z_i$ for all $i = 1, 2, \cdots, N$. That is, each domain D_j is divided into m_j subdivisions D_{ki1}, D_{ki2}, $\cdots$, D_{kim_j} and each partition corresponds to a $D_{1S_1} \times D_{2S_2} \times \cdots \times D_{NS_N}$. In this file, each partition will be randomly assigned to a bucket which means that records within one partition will stay together; a partition will not be split when it is assigned to a bucket. We shall assume here that this size of a partition is not larger than the bucket size.

Consider the following two extreme cases.

Case 1: In this case, $z_i = 1$ for all i. Since $z_i = 1$, each partition contains only one record. Obviously, the partition file thus constructed is a random file because each record will be randomly assigned to buckets.

Case 2: In this case, the partition size is equal to the bucket size. This implies that once a partition is assigned to a bucket, no other partition can occupy this bucket any more. In other words, partitions and buckets have a one-to-one correspondence. A partition file thus constructed is obviously a Cartesian product file.

Since the partition file concept is quite similar to the random file concept, it is not difficult to use the reasoning process used in the later part of Section 3 to derive a formula for the average number of buckets to be examined over all possible partial match queries in a partition file.

Let there be N attributes and let $d_i = m_i z_i$ for all $i = 1, 2, \cdots, N$. Let NP denote the total number of partitions. Let $NPPB$ denote the number of partitions per bucket. Let NQ denote the number of queries. Let NB denote the number of buckets. Let ANB_P denote the average number of buckets to be examined of this partition file over all possible partial match queries. Thus,

$$
ANB_P = (1/NQ) \cdot \left(\sum_{j=1}^{N-1} \Sigma d_{i_1} d_{i_2} \cdots d_{i_j} \cdot NB \right.
$$
$$
\{d_{i_1}, d_{i_2}, \cdots, d_{i_j}\} \in \{d_1, d_2, \cdots, d_N\}
$$
$$
\{m_{i_1}, m_{i_2}, \cdots, m_{i_j}\} \in \{m_1, m_2, \cdots, m_N\}
$$
$$
i_1 < i_2 < \cdots < i_j
$$
$$
\cdot (1 - (C(NP - (NP/m_{i_1} \cdots m_{i_j}), NPPB)/
$$
$$
\left. C(NP, NPPB))) \right) \tag{4.1}
$$

where $NP = m_1 m_2 \cdots m_N$ and $NPPB = NP/NB$.

Let us derive the formulas for random files and Cartesian product files, respectively.

Random Files: For random files, $z_i = 1$ and $m_i = d_i$ for all $i = 1, 2, \cdots, N$. In this case, $NP = m_1 m_2 \cdots m_N = d_1 d_2 \cdots d_N = NR$ (the total number of possible records) and $NPPB = NR/NB = BZ$ (the bucket size). Thus,

$$
ANB_R = (1/NQ)
$$
$$
\cdot \left(\sum_{j=1}^{N-1} \sum_{\substack{\{d_{i_1}, d_{i_2}, \cdots d_{i_j}\} \in \{d_1, d_2, \cdots, d_N\} \\ i_1 < i_2 < \cdots < i_j}} d_{i_1} d_{i_2} \cdots \right.
$$
$$
\cdot\ d_{i_j} \cdot NB \cdot (1 - (C(NR - (NR/d_{i_1} \cdots d_{i_j}), BZ)/
$$
$$
\left. C(NR, BZ))) \right)
$$

which is exactly the same as (3.2).

Cartesian Product Files: For Cartesian product files, $NPPB = 1$. We therefore have

$$
ANB_{CP}
$$
$$
= \frac{1}{NQ} \left(\sum_{j=1}^{N-1} \Sigma d_{i_1} d_{i_2} \cdots d_{i_j} \cdot NB \right.
$$
$$
\{d_{i_1}, d_{i_2}, \cdots, d_{i_j}\} \in \{d_1, d_2, \cdots, d_N\}
$$
$$
\{m_{i_1}, m_{i_2}, \cdots, m_{i_j}\} \in \{m_1, m_2, \cdots, m_N\}
$$
$$
i_1 < i_2 < \cdots < i_j
$$
$$
\left. \cdot (1 - (C(NP - (NP/m_{i_1} m_{i_2} \cdots m_{i_j}), 1)/C(NP, 1))) \right)
$$

$$= \frac{1}{NQ}\left(\sum_{j=1}^{N-1} \Sigma d_{i_1} d_{i_2} \cdots d_{i_j} \cdot NB\right.$$

$$\{d_{i_1}, d_{i_2}, \cdots, d_{i_j}\} \in \{d_1, d_2, \cdots, d_N\}$$

$$\{m_{i_1}, m_{i_2}, \cdots, m_{i_j}\} \in \{m_1, m_2, \cdots, m_N\}$$

$$i_1 < i_2 < \cdots < i_j$$

$$\left.\cdot (1 - (NP(1 - (1/m_{i_1} m_{i_2} \cdots m_{i_j}))/NP))\right)$$

$$= \frac{1}{NQ}\left(\sum_{j=1}^{N-1} \Sigma d_{i_1} d_{i_2} \cdots d_{i_j} \cdot NB/m_{i_1} m_{i_2} \cdots m_{i_j}\right)$$

$$\{d_{i_1}, d_{i_2}, \cdots, d_{i_j}\} \in \{d_1, d_2, \cdots, d_N\}$$

$$\{m_{i_1}, m_{i_2}, \cdots, m_{i_j}\} \in \{m_1, m_2, \cdots, m_N\}$$

$$i_1 < i_2 < \cdots < i_j$$

$$= \frac{NB}{NQ}\left(\sum_{j=1}^{N-1} \Sigma z_{i_1} z_{i_2} \cdots z_{i_j}\right)$$

$$\{z_{i_2}, z_{i_2}, \cdots, z_{i_j}\} \in \{z_1, z_2, \cdots, z_N\}$$

$$i_1 < i_2 < \cdots < i_j$$

which is exactly the same as (3.1).

The fact that random files and Cartesian product files are special cases of partition files is intriguing. It is our feeling that the superiority of Cartesian product files can be established by considering (4.1) directly.

V. Conclusions

In this paper, we have established the superiority of Cartesian product files by proving a theorem showing that the perfomance of Cartesian product files is always better than that of random files. Although we have to impose the restriction at this moment that each domain must be divided into at least two subdivisions and each domain must contain more than one element, this theorem is still significant because those restrictions are quite natural ones. We are working on a new proof which we believe will take care of the special cases.

We have also proposed a new file structure called partition file which is shown to be a general model for both Cartesian product files and random files. That is, both Cartesian product files and random files are special cases of partition files. This model projects new insights into multiattribute file design and we believe that some new and interesting result will be obtained by investigating and studying this new model.

Appendix

Proof of Theorem 3.1

Lemma 1: Let $r \geqslant 2$ and $s \geqslant 2$, where both r and s are positive integers. Then

$$\left(1 - \frac{1}{rs}\right)^{2r-1} < 1 - \frac{r-1}{rs-1}.$$

Proof: We shall prove this inequality by induction. Let

$$r = 2, \quad \left(1 - \frac{1}{rs}\right)^{2r-1} = \left(1 - \frac{1}{2s}\right)^3$$

and

$$1 - \frac{r-1}{rs-1} = 1 - \frac{1}{2s-1}.$$

Therefore,

$$\left(1 - \frac{1}{2s-1}\right) - \left(1 - \frac{1}{2s}\right)^3 = \frac{2s-2}{2s-1} - \frac{8s^3 + 6s - 12s^2 - 1}{8s^3}$$

$$= \frac{16s^4 - 16s^3 - (16s^4 + 12s^2 - 24s^3 - 2s - 8s^3 - 6s + 12s^2 + 1)}{(2s-1) \cdot 8s^3}$$

$$= \frac{16s^3 - 24s^2 + 8s - 1}{(2s-1) \cdot 8s^3}$$

$$= \frac{8s(2s^2 - 3s + 1) - 1}{(2s-1) \cdot 8s^3}$$

$$= \frac{8s(2s-1)(s-1) - 1}{(2s-1) \cdot 8s^3} > 0, \quad \text{for all } s \geqslant 2.$$

Hence

$$\left(1-\frac{1}{rs}\right)^{2r-1} < 1-\frac{r-1}{rs-1}, \quad \text{for } r=2 \text{ and } s \geqslant 2.$$

Suppose $r = k$. The inequality still holds.
That is,

$$\left(1-\frac{1}{ks}\right)^{2k-1} < 1-\frac{k-1}{ks-1} \quad \text{for some } k \geqslant 2 \tag{1}$$

and all $s \geqslant 2$.

Consider the case $r = k+1$ and $s \geqslant 2$:

$$\left(1-\frac{1}{rs}\right)^{2r-1} = \left(1-\frac{1}{(k+1)s}\right)^{2k+1} = \left(1-\frac{1}{ks'}\right)^{2k+1}$$

where

$$s' = \frac{k+1}{k} s > 2.$$

So

$$\left(1-\frac{1}{(k+1)s}\right)^{2k+1} = \left(1-\frac{1}{ks'}\right)^{2k-1} \cdot \left(1-\frac{1}{ks'}\right)^{2}.$$

From (1), we have

$$\left(1-\frac{1}{ks'}\right)^{2k-1} < 1-\frac{k-1}{ks'-1}.$$

Therefore,

$$\left(1-\frac{1}{(k+1)s}\right)^{2k+1} < \left(1-\frac{k-1}{ks'-1}\right)\cdot\left(1-\frac{1}{ks'}\right)^{2} = \left(\frac{ks'-k}{ks'-1}\right)\cdot\left(1-\frac{1}{ks'}\right)^{2} = \left(\frac{(k+1)s-k}{(k+1)s-1}\right)\cdot\left(1-\frac{1}{(k+1)s}\right)^{2}$$

If we can show

$$\left(1-\frac{k}{(k+1)s-1}\right) > \left(\frac{(k+1)s-k}{(k+1)s-1}\right)\cdot\left(1-\frac{1}{(k+1)s}\right)^{2},$$

then

$$1-\frac{k}{(k+1)s-1} > \left(1-\frac{1}{(k+1)s}\right)^{2k+1}$$

is trivial.

In the following, we shall show that

$$1-\frac{k}{(k+1)s-1} > \frac{(k+1)s-k}{(k+1)s-1}\cdot\left(1-\frac{1}{(k+1)s}\right)^{2}\cdot\left(1-\frac{k}{(k+1)s-1}\right) - \left(\left(\frac{(k+1)s-k}{(k+1)s-1}\right)\left(1-\frac{1}{(k+1)s}\right)^{2}\right)$$

$$= \frac{(k+1)s-k-1}{(k+1)s-1} - \frac{(k+1)s-k}{(k+1)s-1}\cdot\left(\frac{(k+1)s-1}{(k+1)s}\right)^{2}$$

$$= \frac{(k+1)s-k-1}{(k+1)s-1} - \frac{(k+1)s-k}{(k+1)s-1}\cdot\left(\frac{(k+1)s-1}{(k+1)s}\right)^{2}$$

$$= \frac{1}{(k+1)s-1}\left(\left(((k+1)s-k)-1\right)-((k+1)s-k)\cdot\left(\frac{(k+1)s-1}{(k+1)s}\right)^{2}\right) = \frac{1}{(k+1)s-1}\left(((k+1)s-k)\cdot\left(1-\left(\frac{(k+1)s-1}{(k+1)s}\right)^{2}\right)-1\right) = \frac{1}{(k+1)s-1}\cdot\left(((k+1)s-k)\cdot\frac{((k+1)s)^2-((k+1)s-1)^2}{((k+1)s)^2}-1\right)$$

$$= \frac{1}{(k+1)s-1}\cdot\left(((k+1)s-k)\cdot\frac{2(k+1)s-1}{(k+1)^2s^2}-1\right)$$

$$= \frac{1}{(k+1)s-1}\cdot\left(\frac{2(k+1)^2s^2+k-(2k+1)(k+1)s}{(k+1)^2s^2}-1\right)$$

$$> \frac{1}{(k+1)s-1}\cdot\left(2-\frac{(2k+1)(k+1)s}{(k+1)^2s^2}-1\right)$$

$$= \frac{1}{(k+1)s-1}\cdot\left(1-\frac{(2k+1)}{(k+1)s}\right).$$

Consider

$$1-\frac{2k+1}{(k+1)s}.$$

Since $s \geqslant 2$,

$$1-\frac{2k+1}{(k+1)s} \geqslant 1-\frac{2k+1}{2(k+1)} = 1-\frac{2k+1}{2k+2} > 0.$$

Hence

$$\left(1-\frac{1}{rs}\right)^{2r-1} < 1-\frac{r-1}{s-1} \text{ holds, } \quad \text{when } r = k+1.$$

Therefore,

$$\left(1-\frac{1}{rs}\right)^{2r-1} < 1-\frac{r-1}{rs-1} \quad \text{for all } r \geqslant 2 \text{ and } s \geqslant 2.$$

This is the proof. Q.E.D.

Lemma 2: Let $A = a \cdot \bar{a}$ and $B = b \cdot \bar{b}$, where $a, \bar{a}, b$, and $\bar{b}$ are all positive integers and are all greater than 1. Then $\bar{a}(C(AB, A)\cdot(\bar{b}-1) - C(AB-ab, A)\cdot\bar{b}) + a(C(AB, A)\cdot(b-1) - C(AB-\bar{a}\bar{b}, A)\cdot b) > 0$.

Proof: Let

$$\Delta = \bar{a}(C(AB,A)\cdot(\bar{b}-1) - C(AB-ab,A)\cdot\bar{b}) + a(C(AB,A)\cdot(b-1) - C(AB-\bar{a}\bar{b},A)\cdot b)\cdot$$

$$\begin{aligned}\Delta &= C(AB,A)(\bar{a}\bar{b}+ab-\bar{a}-a) - C(AB-ab,A)\bar{a}\bar{b} - C(AB-\bar{a}\bar{b},A)ab \\ &= \frac{(AB)!}{(AB-A)!A!}(\bar{a}\bar{b}+ab-\bar{a}-a) - \frac{(AB-ab)!}{(AB-ab-A)!A!}\cdot\bar{a}\bar{b} - \frac{(AB-\bar{a}\bar{b})!}{(AB-\bar{a}\bar{b}-A)!\cdot A!}\cdot ab \\ &\frac{1}{(AB-A)!A!(AB-ab-A)!(AB-\bar{a}\bar{b}-A)!} \\ &\quad\cdot((AB)!\cdot(AB-ab-A)!\cdot(AB-\bar{a}\bar{b}-A)! \\ &\quad\cdot(\bar{a}\bar{b}+ab-\bar{a}-a) - ((AB-ab)!\cdot(AB-A)! \\ &\quad\cdot(AB-\bar{a}\bar{b}-A)!\bar{a}\bar{b} + (AB-\bar{a}\bar{b})!(AB-A)! \\ &\quad\cdot(AB-ab-A)!\cdot ab)) \\ &= \frac{1}{(AB-A)!A!(AB-ab-A)!(AB-\bar{a}\bar{b}-A)!}\cdot\Delta'\end{aligned}$$

where

$$\begin{aligned}\Delta' &= (AB)!(AB-ab-A)!(AB-\bar{a}\bar{b}-A)!(\bar{a}\bar{b}+ab-\bar{a}-a) \\ &\quad - ((AB-ab)!(AB-A)!(AB-\bar{a}\bar{b}-A)!\bar{a}\bar{b} \\ &\quad + (AB-\bar{a}\bar{b})!\cdot(AB-A)!(AB-ab-A)!ab) \\ &= (AB)!(AB-ab-A)!(AB-\bar{a}\bar{b}-A)! \\ &\quad\cdot(\bar{a}\bar{b}+ab-\bar{a}-a) - ((AB-A)!(AB-ab-A)! \\ &\quad\cdot(AB-\bar{a}\bar{b}-A)!((AB-ab)\cdot(AB-ab-1) \\ &\quad\cdots(AB-ab-A+1)\cdot\bar{a}\bar{b} + (AB-\bar{a}\bar{b})(AB-\bar{a}\bar{b}-1) \\ &\quad\cdots(AB-\bar{a}\bar{b}-A+1)\cdot ab)) \\ &= (AB-A)!(AB-ab-A)!(AB-\bar{a}\bar{b}-A)! \\ &\quad\cdot(AB\cdot(AB-1)\cdots(AB-A+1)\cdot(\bar{a}\bar{b}+ab-\bar{a}-a) \\ &\quad-(AB-ab)(AB-ab-1)\cdots(AB-ab-A+1) \\ &\quad\cdot\bar{a}\bar{b} - (AB-\bar{a}\bar{b})(AB-\bar{a}\bar{b}-1)\cdots(AB-\bar{a}\bar{b}-A+1) \\ &\quad\cdot ab) = (AB-A)!(AB-ab-A)!(AB-\bar{a}\bar{b}-A)!\cdot\Delta'',\end{aligned}$$

$$\begin{aligned}\Delta'' &= AB\cdot(AB-1)\cdots(AB-A+1)\cdot(\bar{a}\bar{b}+ab-\bar{a}-a) \\ &\quad-(AB-ab)(AB-ab-1)\cdots(AB-ab-A+1) \\ &\quad\cdot\bar{a}\bar{b}-(AB-\bar{a}\bar{b})(AB-\bar{a}\bar{b}-1)\cdots(AB-\bar{a}\bar{b}-A+1) \\ &\quad\cdot ab = AB(AB-1)\cdots(AB-A+1)(\bar{a}\bar{b}+ab-\bar{a}-a) \\ &\quad-AB(\bar{a}\bar{b}-1)(AB-ab-1)\cdots(AB-ab-A+1) \\ &\quad-AB(ab-1)(AB-\bar{a}\bar{b}-1)\cdots(AB-\bar{a}\bar{b}-A+1) \\ &= AB\cdot((AB-1)\cdots(AB-A+1)\cdot(\bar{a}\bar{b}+ab-\bar{a}-a) \\ &\quad-(AB-ab-1)\cdots(AB-ab-A+1)\cdot(\bar{a}\bar{b}-1) \\ &\quad-(AB-\bar{a}\bar{b}-1)\cdots(AB-\bar{a}\bar{b}-A+1)\cdot(ab-1)) \\ &= AB\cdot\Delta'''\end{aligned}$$

and

$$\begin{aligned}\Delta''' &= (AB-1)(AB-2)\cdots(AB-A+1)(\bar{a}\bar{b}+ab-\bar{a}-a) \\ &\quad-(AB-ab-1)\cdots(AB-ab-A+1)\cdot(\bar{a}\bar{b}-1) \\ &\quad-(AB-\bar{a}\bar{b}-1)\cdots(AB-\bar{a}\bar{b}-A+1)\cdot(ab-1).\end{aligned}$$

Now our problem is to show that Δ''' is always positive. Take

$$\alpha = \frac{(AB-ab-1)(AB-ab-2)\cdots(AB-ab-A+1)}{(AB-1)(AB-2)\cdots(AB-A+1)} < 1$$

and

$$\beta = \frac{(AB-\bar{a}\bar{b}-1)(AB-\bar{a}\bar{b}-2)\cdots(AB-\bar{a}\bar{b}-A+1)}{(AB-1)(AB-2)\cdots(AB-A+1)} < 1.$$

So

$$\begin{aligned}\Delta''' &= (AB-1)(AB-2)\cdots(AB-A+1)(\bar{a}\bar{b}+ab-\bar{a}-a) \\ &\quad-(\bar{a}\bar{b}-1)\alpha(AB-1)(AB-2)\cdots(AB-A+1) \\ &\quad-(ab-1)\beta(AB-1)(AB-2)\cdots(AB-A+1) \\ &= (AB-1)(AB-2)\cdots(AB-A+1)\cdot((\bar{a}\bar{b}+ab \\ &\quad-\bar{a}-a)-\alpha(\bar{a}\bar{b}-1)-\beta(ab-1)) \\ &= (AB-1)(AB-2)\cdots(AB-A+1)\cdot\eta\end{aligned}$$

where

$$\begin{aligned}\eta &= (\bar{a}\bar{b}+ab-\bar{a}-a)-\alpha(\bar{a}\bar{b}-1)-\beta(ab-1) \\ &= ((\bar{a}\bar{b}-1)+(1-\bar{a})+(ab-1)+(1-a)) \\ &\quad-\alpha(\bar{a}\bar{b}-1)-\beta(ab-1) \\ &= (1-\alpha)(\bar{a}\bar{b}-1)+(1-\bar{a})+(1-\beta)(ab-1)+(1-a).\end{aligned}$$

If $\alpha < 1-(\bar{a}-1)/(\bar{a}\bar{b}-1)$ and $\beta < 1-(a-1)/(ab-1)$ simultaneously, then η is positive and so is Δ'''.

Now our problem is again to examine the inequality as following:

$$\alpha < 1 - \frac{\bar{a}-1}{\bar{a}\bar{b}-1} \text{ and } \beta < 1 - \frac{a-1}{ab-1}.$$

Since

$$\alpha = \frac{(AB-ab-1)(AB-ab-2)\cdots(AB-ab-A+1)}{(AB-1)(AB-2)\cdots(AB-A+1)}$$

$$< \frac{AB-ab}{AB} \cdot \frac{AB-ab}{AB} \cdots \frac{AB-ab}{AB}$$

$$= \left(1 - \frac{ab}{AB}\right)^{A-1} = \left(1 - \frac{1}{\bar{a}\bar{b}}\right)^{A-1},$$

$$\beta = \frac{(AB-\bar{a}\bar{b}-1)(AB-\bar{a}\bar{b}-2)\cdots(AB-\bar{a}\bar{b}-A+1)}{(AB-1)(AB-2)\cdots(AB-A+1)}$$

$$< \frac{AB-\bar{a}\bar{b}}{AB} \cdot \frac{AB-\bar{a}\bar{b}}{AB} \cdots \frac{AB-\bar{a}\bar{b}}{AB}$$

$$= \left(1 - \frac{\bar{a}\bar{b}}{AB}\right)^{A-1} = \left(1 - \frac{1}{ab}\right)^{A-1}$$

and $a, \bar{a}, b$, and $\bar{b} \geqslant 2$, we have $A \geqslant 2a$ and $A \geqslant 2\bar{a}$. So

$$\left(1 - \frac{1}{\bar{a}\bar{b}}\right)^{A-1} \leqslant \left(1 - \frac{1}{\bar{a}\bar{b}}\right)^{2\bar{a}-1}$$

and

$$\left(1 - \frac{1}{ab}\right)^{A-1} \leqslant \left(1 - \frac{1}{ab}\right)^{2a-1}.$$

Because

$$\alpha < \left(1 - \frac{1}{\bar{a}\bar{b}}\right)^{A-1}$$

and

$$\beta < \left(1 - \frac{1}{ab}\right)^{A-1},$$

we have

$$\alpha < \left(1 - \frac{1}{\bar{a}\bar{b}}\right)^{2\bar{a}-1}$$

and

$$\beta < \left(1 - \frac{1}{ab}\right)^{2a-1}.$$

By Lemma 1,

$$\left(1 - \frac{1}{\bar{a}\bar{b}}\right)^{2\bar{a}-1} < 1 - \frac{\bar{a}-1}{\bar{a}\bar{b}-1}$$

and

$$\left(1 - \frac{1}{ab}\right)^{2a-1} < 1 - \frac{a-1}{ab-1}.$$

Hence,

$$\alpha < 1 - \frac{\bar{a}-1}{\bar{a}\bar{b}-1}$$

and

$$\beta < 1 - \frac{a-1}{ab-1}.$$

These imply that η is positive and so are Δ''', Δ'', Δ' and Δ. Q.E.D

Theorem 3.1: Let there be N attributes where the domain size of each attribute is d_i. Let the number of buckets be NB. Let the Cartesian product file partition each d_i into m_i subdivisions. That is, $d_i = m_i z_i$. Then $ANB_{CP} < ANB_R$ if $m_i \geqslant 2$ and $z_i \geqslant 2$ for all $i = 1, 2, \cdots, N$.

Proof:

$$ANB_R = \frac{NB}{NQ}\left(NQ - (NR/C(NR, BZ)) \cdot \left(\sum_{j=1}^{N-1} \underset{\substack{\{d_{i_1}, d_{i_2}, \cdots, d_{i_j}\} \in \{d_1, d_2, \cdots, d_N\} \\ i_1 < i_2 < \cdots < i_j}}{\Sigma} (1/d_{i_1}d_{i_2}\cdots d_{i_j}) \cdot C(NR - d_{i_1}d_{i_2}\cdots d_{i_j}, BZ)\right)\right)$$

and

$$ANB_{CP} = \frac{NB}{NQ}\left(\sum_{j=1}^{N-1} \underset{\substack{\{z_{i_1}, z_{i_2}, \cdots, z_{i_j}\} \in \{z_1, z_2, \cdots, z_N\} \\ i_1 < i_2 < \cdots < i_j}}{\Sigma} z_{i_1}z_{i_2}\cdots z_{i_j}\right),$$

where

$$NR = d_1 d_2 \cdots d_N,\ BZ = z_1 z_2 \cdots z_N,\ NB = NR/BZ$$

and

$$NQ = \sum_{j=1}^{N-1} \underset{\substack{\{d_{i_1}, d_{i_2}, \cdots, d_{i_j}\} \in \{d_1, d_2, \cdots, d_N\} \\ i_1 < i_2 < \cdots < i_j}}{\Sigma} d_{i_1}d_{i_2}\cdots d_{i_j}.$$

Let $\delta = ANB_R - ANB_{CP}$. We want to show that $\delta > 0$:

$$\delta = \frac{NB}{NQ}\left(\left(NQ - \left((NR/C(NR, BZ)) \cdot \left(\sum_{j=1}^{N-1} \underset{\substack{\{d_{i_1}, d_{i_2}, \cdots, d_{i_j}\} \in \{d_1, d_2, \cdots, d_N\} \\ i_1 < i_2 < \cdots < i_j}}{\Sigma} (1/d_{i_1}d_{i_2}\cdots d_{i_j}) \cdot C(NR - d_{i_1}d_{i_2}\cdots d_{i_j}, BZ)\right)\right)\right) - \sum_{j=1}^{N-1} \underset{\substack{\{z_{i_1}, z_{i_2}, \cdots, z_{i_j}\} \in \{z_1, z_2, \cdots, z_N\} \\ i_1 < i_2 < \cdots < i_j}}{\Sigma} z_{i_1}z_{i_2}\cdots z_{i_j}\right)$$

$$= \frac{NB}{NQ} \cdot \delta'$$

where

$$\delta' = NQ - \left(\frac{NR}{C(NR, BZ)}\right.$$

$$\cdot\left(\sum_{j=1}^{N-1} \sum_{\substack{\{d_{i_1}, d_{i_2}, d_{i_j}\} \in \{d_1, d_2 \cdots, d_N\} \\ i_1 < i_2 < \cdots < i_j}} \frac{1}{d_{i_1} d_{i_2} \cdots d_{i_j}}\right.$$

$$\left.\left.\cdot\, C(NR - d_{i_1} d_{i_2} \cdots d_{i_j}, BZ)\right)\right)$$

$$-\left(\sum_{j=1}^{N-1} \Sigma z_{i_1} z_{i_2} \cdots z_{i_j}\right)\{z_{i_1}, z_{i_2}, \cdots, z_{i_j}\}$$

$$\in \{z_1, z_2, \cdots, z_N\} i_1 < i_2 < \cdots < i_j.$$

Let $d_i = z_i \cdot m_i$.

$$\delta' = \left(\sum_{j=1}^{N-1} \sum_{\substack{\{z_{i_1}, z_{i_2}, \cdots, z_{i_j}\} \in \{z_1, z_2, \cdots, z_N\} \\ \{m_{i_1}, m_{i_2}, \cdots, m_{i_j}\} \in \{m_1, m_2, \cdots, m_N\} \\ i_1 < i_2 < \cdots < i_j}} z_{i_1} z_{i_2} \cdots z_{i_j}\right.$$

$$\left.\cdot\, m_{i_1} m_{i_2} \cdots m_{i_j}\right) - \frac{z_1 z_2 \cdots z_N m_1 m_2 \cdots m_N}{C(z_1 z_2 \cdots z_N m_1 m_2 \cdots m_N, z_1 z_2 \cdots z_N)}$$

$$\cdot\left(\sum_{j=1}^{N-1} \sum_{\substack{\{z_{i_1}, z_{i_2}, z_{i_j}\} \in \{z_1, z_2, \cdots, z_N\} \\ \{m_{i_1}, m_{i_2}, \cdots, m_{i_j}\} \in \{m_1, m_2, \cdots, m_N\} \\ i_1 < i_2 < \cdots < i_j}}\right.$$

$$\cdot\, \frac{1}{z_{i_1} z_{i_2} \cdots z_{i_j} m_{i_1} m_{i_2} \cdots m_{i_j}}$$

$$\cdot\, C(z_1 z_2 \cdots z_M m_1 m_2 \cdots m_N$$

$$\left.\cdot\, z_{i_1} z_{i_2} \cdots z_{i_j} m_{i_1} m_{i_2} \cdots m_{i_j}, z_1 z_2 \cdots z_N)\right)$$

$$-\left(\sum_{j=1}^{N-1} \Sigma z_{i_1} z_{i_2} \cdots z_{ij}\right)\{z_{i_1}, z_{i_2}, \cdots, z_{i_j}\}$$

$$\in \{z_1, z_2, \cdots, z_N\} i_2 < i_2 < \cdots < i_j$$

$$\delta' = \left(\sum_{j=1}^{N-1} \sum_{\substack{\{z_{i_1}, z_{i_2}, \cdots, z_{i_j}\} \in \{z_1, z_2, \cdots, z_N\} \\ \{m_{i_1}, m_{i_2}, \cdots, m_{i_j}\} \in \{m_1, m_2, \cdots m_N\} \\ i_1 < i_2 < \cdots < i_j}} z_{i_1} z_{i_2} \cdots\right.$$

$$\left.\cdot\, z_{i_j}(m_{i_1} m_{i_2} \cdots m_{i_j} - 1)\right)$$

$$-\frac{z_1 z_2 \cdots z_N m_1 m_2 \cdots m_N}{C(z_1 z_2 \cdots z_N m_1 m_2 \cdots m_N, z_1 z_2, \cdots z_N)}$$

$$\cdot\left(\sum_{j=1}^{N-1} \sum_{\substack{\{z_{i_1}, z_{i_2}, \cdots, z_{i_j}\} \in \{z_1, z_2, \cdots, z_N\} - \\ \{m_{i_1}, m_{i_2}, \cdots, m_{i_j}\} \in \{m_1, m_1, \cdots, m_N\} \\ i_1 < i_2 < \cdots < i_j}}\right.$$

$$\cdot\left(\frac{1}{z_{i_1} z_{i_2} \cdots z_{i_j} m_{i_1} m_{i_2} \cdots m_{i_j}}\right.$$

$$\cdot\, C(z_1 z_2 \cdots z_N m_1 m_2 \cdots m_N z_{i_1} z_{i_2} \cdots z_{i_j}$$

$$\left.\left.\cdot\, m_{i_1} m_{i_2} \cdots m_{i_j}, z_1 z_2 \cdots z_N)\right)\right).$$

$$\delta' = \frac{z_1 z_2 \cdots z_N m_1 m_2 \cdots m_N}{C(z_1 z_2 \cdots z_N m_1 m_2 \cdots m_N, z_1 z_2 \cdots z_N)}$$

$$\cdot\left(\sum_{j=1}^{N-1} \Sigma\left(\frac{C(z_1 z_2 \cdots z_N m_1 m_2 \cdots m_N, z_1 z_2 \cdots z_N)}{z_1 z_2 \cdots z_N m_1 m_2 \cdots m_N}\right.\right.$$

$$\{z_{i_1}, z_{i_2}, \cdots, z_{i_j}\} \in \{z_1, z_2, \cdots, z_N\}$$

$$\{m_{i_1}, m_{i_2}, \cdots, m_{i_j}\} \in \{m_1, m_2, \cdots, m_N\}$$

$$i_1 < i_2 < \cdots < i_j$$

$$\cdot\, z_{i_1} z_{i_2} \cdots z_{i_j} \cdot (m_{i_1} m_{i_2} \ldots m_{i_j} - 1)$$

$$- C(z_1 z_2 \cdots z_N m_1 m_2 \cdots m_N - z_{i_1} z_{i_2} \cdots z_{i_j} m_{i_1} m_{i_2}$$

$$\left.\left.\cdots m_{i_j}, z_1 z_2 \cdots z_N)/z_{i_1} z_{i_2} \cdots z_{i_j} m_{i_1} m_{i_2} \cdots m_{i_j}\right)\right)$$

$$= \frac{z_1 z_2 \cdots z_N m_1 m_2 \cdots m_N}{C(z_1 z_2 \cdots z_N m_1 m_2 \cdots m_N, z_1 z_2 \cdots z_N)} \cdot \delta''$$

where

$$\delta'' = \frac{1}{z_1z_2\cdots z_Nm_1m_2\cdots m_N} \cdot \Bigg(\sum_{j=1}^{N-1} \sum_{\substack{\{z_{i_1},z_{i_2},\cdots,z_{i_j}\}\in\{z_1,z_2,\cdots,z_N\}\\ \{m_{i_1},m_{i_2},\cdots,m_{i_j}\}\in\{m_1,m_2,\cdots,m_N\}\\ i_1<i_2<\cdots<i_j}} (C(z_1z_2\cdots z_Nm_1m_2\cdots m_N, z_1z_2\cdots z_N)z_{i_1}z_{i_2}\cdots z_{i_j} \cdot (m_{i_1}m_{i_2}\cdots m_{i_j}-1) - C(z_1z_2\cdots z_Nm_1m_2\cdots m_N - z_{i_1}z_{i_2}\cdots z_{i_j}m_{i_1}m_{i_2}\cdots m_{i_j}, z_1z_2\cdots z_N)\cdot z_{i_{j+1}}z_{i_{j+2}}\cdots z_{i_N} \cdot m_{i_{j+1}}m_{i_{j+2}}\cdots m_{i_N})\Bigg)$$

$$= \frac{1}{z_1z_2\cdots z_Nm_1m_2\cdots m_N}\delta'''$$

and

$$\delta''' = \sum_{j=1}^{N-1} \sum_{\substack{\{z_{i_1},z_{i_2},\cdots,z_{i_j}\}\in\{z_1,z_2,\cdots,z_N\}\\ \{m_{i_1},m_{i_2},\cdots,m_{i_j}\}\in\{m_1,m_2,\cdots,m_N\}\\ i_1<i_2<\cdots<i_j}} (((C(z_1z_2\cdots z_Nm_1m_2\cdots m_N, z_1z_2\cdots z_N)z_{i_1}z_{i_2}\cdots z_{i_j}\} \cdot (m_{i_1}m_{i_2}\cdots m_{i_j}-1)) - C(z_1z_2\cdots z_Nm_1m_2\cdots m_N - z_{i_{j+1}}z_{i_{j+2}}\cdots z_{i_N}m_{i_{j+1}}m_{i_{j+2}}\cdots m_{i_N}, z_1z_2\cdots z_N) \cdot z_{i_1}z_{i_2}\cdots z_{i_j}\cdot m_{i_1}m_{i_2}\cdots m_{i_j})$$

$$= \sum_{j=1}^{N-1} \sum_{\substack{\{z_{i_1},z_{i_2},\cdots,z_{i_j}\}\in\{z_1,z_2,\cdots,z_N\}\\ \{m_{i_1},m_{i_2},\cdots,m_{i_j}\}\in\{m_1,m_2,\cdots,m_N\}\\ i_1<i_2<\cdots<i_j}} (z_{i_1}z_{i_2}\cdots z_{i_j})(C(z_1z_2\cdots z_Nm_1m_2\cdots m_N, z_1z_2\cdots z_N) \cdot (m_{i_1}m_{i_2}\cdots m_{i_j}-1) - C(z_1z_2\cdots z_Nm_1m_2\cdots m_N - z_{i_{j+1}}z_{i_{j+2}}\cdots z_{i_N}m_{i_{j+1}}m_{i_{j+2}}\cdots m_{i_N}, z_1z_2\cdots z_N)\cdot m_{i_1}m_{i_2}\cdots m_{i_j}).$$

Let

$$a_{(i_1,i_2,\cdots,i_j)} = z_{i_1}z_{i_2}\cdots z_{i_j}$$

and

$$\bar{a}_{(i_1,i_2,\cdots,i_j)} = \frac{z_1z_2\cdots z_N}{z_{i_1}z_{i_2}\cdots z_{i_j}} = z_{i_{j+1}}z_{i_{j+2}}\cdots z_{i_N}.$$

Let

$$A = z_1z_2\cdots z_N.$$

Similarly,

$$b_{(i_1,i_2,\cdots,i_j)} = m_{i_1}m_{i_2}\cdots m_{i_j}$$

$$\bar{b}_{(i_1,i_2,\cdots,i_j)} = \frac{m_1m_2\cdots m_N}{m_{i_1}m_{i_2}\cdots m_{i_j}} = m_{i_{j+1}}m_{i_{j+2}}\cdots m_{i_N}.$$

and $B = m_1m_2\cdots m_N$.

Then, δ''' can be rewritten as the following:

$$\delta''' = \sum_{j=1}^{N-1} \Sigma a_{(i_1,i_2,\cdots,i_j)}(C(AB,A)(b_{(i_1,i_2,\cdots,i_j)}-1) - C(AB - \bar{a}_{(i_1,i_2,\cdots,i_j)}\cdot \bar{b}_{(i_1,i_2,\cdots,i_j)}, A) \cdot b_{(i_1,i_2,\cdots,i_j)}).$$

Since

$$a_{(i_1,i_2,\cdots,i_j)}(C(AB,A)\cdot(b_{(i_1,i_2,\cdots,i_j)}-1) - C(AB-\bar{a}_{(i_1,i_2,\cdots,i_j)}\cdot\bar{b}_{(i_1,i_2,\cdots,i_j)},A)\cdot b_{(i_1,i_2,\cdots,i_j)})$$

and

$$\bar{a}_{(i_1,i_2,\cdots,i_j)}(C(AB,A)\cdot(\bar{b}_{(i_1,i_2,\cdots,i_j)}-1) - C(AB-a_{(i_1,i_2,\cdots,i_j)}b_{(i_1,i_2,\cdots,i_j)},A)\cdot\bar{b}_{(i_1,i_2,\cdots,i_j)})$$

in δ''' appears in pair, if we can prove that the value of

$$a_{(i_1,i_2,\cdots,i_j)} (C(AB,A)(b_{(i_1,i_2,\cdots,i_j)}-1) - C(AB-\bar{a}_{(i_1,i_2,\cdots,i_j)}\cdot\bar{b}_{(i_1,i_2,\cdots,i_j)},A)\cdot b_{(i_1,i_2,\cdots,i_j)}) + \bar{a}_{(i_1,i_2,\cdots,i_j)}\cdot(C(AB,A)\cdot(\bar{b}_{(i_1,i_2,\cdots,i_j)}-1) - C(AB-a_{(i_1,i_2,\cdots,i_j)}b_{(i_1,i_2,\cdots,i_j)},A).$$

$\bar{b}_{(i_1,i_2,\cdots,i_j)}$ is always positive, then we can conclude $\delta''' > 0$.

By Lemma 2, since $A = a\cdot\bar{a}$ and $B = b\cdot\bar{b}$, $(a, \bar{a}, b, \bar{b} \in N$ and $a, \bar{a}, b, \bar{b} > 1)$ imply $\bar{a}(C(AB, A)\cdot(\bar{b}-1) - C(AB - ab, A)\cdot\bar{b}) + a(C(AB, A)\cdot(b-1) - C(AB - \bar{a}\bar{b}, A)\cdot b) > 0$.

Note that

$$a_{(i_1,i_2,\cdots,i_j)}, \bar{a}_{(i_1,i_2,\cdots,i_j)}, b_{(i_1,i_2,\cdots,i_j)}, \bar{b}_{(i_1,i_2,\cdots,i_j)} \in N$$

and they are all greater than 1. Let

$$A = a_{(i_1,i_2,\cdots,i_j)}\cdot\bar{a}_{(i_1,i_2,\cdots,i_j)}$$

and

$$B = b_{(i_1,i_2,\cdots,i_j)} \cdot \bar{b}_{(i_1,i_2,\cdots,i_j)}.$$

We have

$$\begin{aligned} &a_{(i_1,i_2,\cdots,i_j)}(C(AB,A) \cdot (b_{(i_1,i_2,\cdots,i_j)} - 1) \\ &\quad - C(AB - \bar{a}_{(i_1,i_2,\cdots,i_j)} \cdot \bar{b}_{(i_1,i_2,\cdots,i_j)}, A) \\ &\quad \cdot b_{(i_1,i_2,\cdots,i_j)}) + \bar{a}_{(i_1,i_2,\cdots,i_j)} \cdot (C(AB,A) \\ &\quad \cdot (\bar{b}_{(i_1,i_2,\cdots,i_j)} - 1) - C(AB - a_{(i_1,i_2,\cdots,i_j)} \\ &\quad \cdot b_{(i_1,i_2,\cdots,i_j)}, A) \cdot \bar{b}_{(i_1,i_2,\cdots,i_j)}) > 0. \end{aligned}$$

Hence $\delta''' > 0$ and so are δ'', δ', and δ. We have the proof.

Q.E.D

References

[1] A. V. Aho and J. D. Ullman, "Optimal partial match retrieval when fields are independently specified," *ACM Trans. Database Syst.*, vol. 4, pp. 168–179, June 1979.

[2] J. L. Bentley and J. H. Friedman, "Data structures for range searching," *Comput. Surveys*, vol. 11, pp. 397–409, Dec. 1979.

[3] C. C. Chang, R. C. T. Lee, and H. C. Du, "Some properties of Cartesian product files," in *Proc. ACM-SIGMOD 1980 Conf.*, Santa Monica, CA, May 1980, pp. 157–168.

[4] C. C. Chang, R. C. T. Lee, and M. W. Du, "Symbolic gray code as a perfect multiattribute hashing scheme for partial match queries," *IEEE Trans. Software Eng.*, vol. SE-8, May 1982.

[5] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: Freeman, 1979.

[6] R. C. T. Lee, "Clustering analysis and its applications," in *Advances in Information Systems Science*, vol. 8, J. T. Tou, Ed. New York: Plenum, 1981, pp. 169–287.

[7] R. C. T. Lee and S. H. Tseng, "Multi-key sorting," *Policy Anal. Inform. Syst.*, vol. 3, pp. 1–20, Dec. 1979.

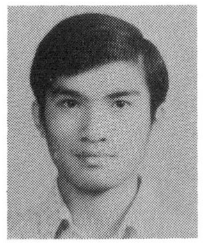
[8] W. C. Lin, R. C. T. Lee, and H. C. Du, "Common properties of some multiattribute file systems," *IEEE Trans. Software Eng.*, vol. SE-5, pp. 160–174, Mar. 1979.

[9] J. H. Liou and S. B. Yao, "Multi-dimensional clustering for data base organizations," *Inform. Syst.*, vol. 2, pp. 187–198, 1977.

[10] R. L. Rivest, "Partial match retrieval algorithms," *SIAM J. Comput.*, vol. 15, pp. 19–50, Mar. 1976.

[11] J. B. Rothnie and T. Lozano, "Attribute based file organization in a paged memory environment," *Commun. Ass. Comput. Mach.*, vol. 17, pp. 63–69, Feb. 1974.

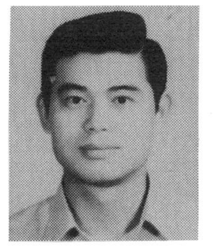
[12] C. Y. Tang, "On the complexity of some file design problems," M.S. thesis, Inst. Comput. and Decision Sci., Nat. Tsing Hua Univ., Hsinchu, Taiwan, Republic of China, 1982.



C. C. Chang was born in Taiwan, Republic of China, in 1954. He received the B.S. degree in applied mathematics and the M.S. degree in computer science from the National Tsing Hua University, Hsinchu, Taiwan, in 1977 and 1979, respectively, and the Ph.D. degree in computer engineering from the National Chiao-Tung University, Hsinchu, Taiwan, in 1982.

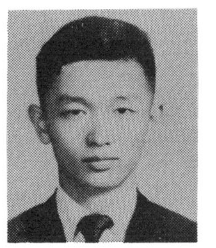
He is presently an Associate Professor of the Institute of Computer Engineering, National Chiao-Tung University. His main research interests include database management system design, computer performance evaluation, and algorithm design.

Dr. Chang is a member of Phi Tau Phi.



M. W. Du (S'70–M'72) received the B.S.E.E. degree from the National Taiwan University in 1966 and the Ph.D. degree from The Johns Hopkins University, Baltimore, MD, in 1972.

He is currently the Director of the Institute of Computer Engineering, National Chiao-Tung University, Hsinchu, Taiwan. His research interests include fault diagnosis, automata theory, algorithm design and analysis, database design, and Chinese I/O design.



R. C. T. Lee was born in Shanghai, China, in 1939. He received the B.S. degree in electrical engineering from the National Taiwan University in 1961 and the M.S. degree in electrical engineering and the Ph.D. degree from the University of California, Berkeley, in 1963 and 1967, respectively.

He is currently the Director of the Institute of Computer and Decision Sciences, National Tsing Hua University, Hsinchu, Taiwan. He previously worked for NCR (California), National Institutes of Health (Bethesda, MD), and the Naval Research Laboratory (Washington, DC) before joining the National Tsing Hua University in 1975. He is the coauthor of *Symbolic Logic and Mechanical Theorem Proving* (New York: Academic), which has been translated into both Japanese and Russian. His research in clustering analysis appeared as a chapter entitled "Clustering Analysis and its Applications" in *Advances in Information Systems Science* (New York: Plenum). He also wrote a chapter on compiler writing for *Handbook of Software Engineering* (New York: Van Nostrand Reinhold). He is the author of more than 50 papers on mechanical theorem proving, database design, and pattern recognition.