SIAM J. COMPUT.
Vol. 12, No. 4, November 1983

0097-5397/83/1204-0010 $01.25/0

CONVEX CLUSTERS IN A DISCRETE m-DIMENSIONAL SPACE*

JOHN L. PFALTZ†

Abstract. A simple procedure is used to dynamically create convex clusters of items in a discrete m-dimensional space. Systems of difference equations are derived that describe the behavior of this cluster formation under the assumption that individual clusters are either of bounded or unbounded size. These equations are used to calculate the expected number and size of such clusters given the number n of items.

For an actual implemented database access and retrieval method, these results provide a way of determining both the expected storage overhead and the expected retrieval costs.

Key words. attribute space, cluster, database, dynamic item entry, retrieval cost, file organization, indexed-descriptor, partial-match retrieval, storage overhead

1. Introduction.

In this paper we will consider the creation of convex clusters formed by the random placement of items in a discrete m-dimensional space with finite bounds. That is, items will be associated with points of the space that may be denoted by m-tuples of integers $(i_1, i_2, \cdots, i_m)$, where $1 \leqq i_i \leqq w_i$. Here w_i denotes the upper bound on that coordinate of the space. Two questions are of particular interest. Given a clustering process and n items in the space:

a. What is the expected number of convex clusters?

b. What is the expected number of convex clusters containing precisely k items?

Although we have deliberately cast these problems in purely mathematical terms, for instance "convex cluster" and "m-dimensional space", they have been motivated by a very practical application, that of determining the expected retrieval cost and expected storage overhead of an efficient information retrieval organization for very large data files [8]. In this interpretation, cells of the m-dimensional space may be viewed as buckets in an m-dimensional attribute space; items may be regarded as data records; and clusters may be blocks in a data file. We will discuss these computer applications more fully in §§ 5 and 8.

2. Descriptors and convex clusters.

Since every cell[1] of the space can be identified by an m-tuple $(i_1, i_2, \cdots, i_m)$ of integers, we can equivalently identify the cell by a *descriptor* consisting of m distinct bit strings called *fields*. In the jth field (of width w_i bits) only the ith bit is set to 1. Thus

00100000 000001 0000100000 00000010

is a 4-field descriptor for the cell (3, 6, 5, 7) in a finite 4-dimensional space consisting of $8 \times 6 \times 10 \times 8 = 3840$ cells. The use of bit descriptors instead of integer m-tuples will at first seem awkward. But, in fact, this transformation simplifies the following analysis.

Every cell in the space is, by itself, a convex subset. Although other definitions of "convexity" in a discrete space are possible, we will adopt the convention that a subset is *convex* if and only if it is a $b_1 \times b_2 \times \cdots \times b_m$-cube. Thus in the 2-dimensional 15×15 space shown in Fig. 1(a) (which is deliberately "small" for illustrative purposes;

* Received by the editors March 6, 1981, and in final revised form August 9, 1982. This research was supported in part by the National Science Foundation under grant MCS80-17779.

† Department of Applied Math. and Computer Science, University of Virginia, Charlottesville, Virginia 22903.

[1] We could equally well speak of "points" in the space. However, the term "cell" seems preferable because we will be associating data items with such points in the space, and "cell" suggests that several such items can be associated with (or stored in) that "cell".

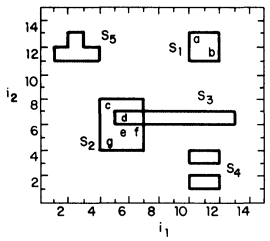

FIG. 1(a)

```
D1  0000000000011000 0000000000001100
D2  0000111000000000 0000111100000000
D3  0000011111111100 0000001000000000
D4  0000000000011000 0101000000000000
```

FIG. 1(b)

```
Da  0000000000010000 0000000000000100
Db  0000000000001000 0000000000001000
    0000000000011000 0000000000001100  = Da ∨ Db = D1

Dc  0000100000000000 0000000100000000
Dd  0000010000000000 0000001000000000
De  0000001000000000 0000001000000000
Df  0000001000000000 0000001000000000
Dg  0000100000000000 0000100000000000
    0000111000000000 0000111100000000  = Dc ∨ Dd ∨ De ∨ Df ∨ Dg
                                       = D2
```

FIG. 1(c)

normally $m > 2$) subsets S_1, S_2 and S_3 are convex subsets, while S_4 and S_5 are not. For convex subsets one can form a *descriptor* by simply OR-ing the descriptors of each of its cells. The descriptors of these convex $b_1 \times b_2 \times \cdots \times b_m$-cubes are characterized by the fact that each field has b_i *consecutive* 1-bits set. Such convex subsets are called *clusters*. Their associated descriptors are *convex cluster descriptors*, or just cluster descriptors. D_1, D_2 and D_3 in Fig. 1(b) are convex descriptors associated with the convex subsets S_1, S_2 and S_3 respectively. D_4 is the descriptor associated with S_4; but neither is convex. Note that there is no adequate descriptor for the set S_5. By the *extent* of a cluster, we mean the number of cells in the space comprising it; that is extent $= b_1 \times b_2 \times \cdots \times b_m$. (One can use "volume" as a synonym for extent.) In Fig. 1, extent $(S_2) = 4$, extent $(S_2) = 12$, and extent $(S_3) = 8$.

Let S_1 and S_2 denote any convex subsets of a space, so that their corresponding descriptors D_1 and D_2 are also convex. Clearly $S_1 \cap S_2$ is convex (or void) and $D_1 \wedge D_2$, if it exists, must be the convex descriptor of that intersection set. (To "exist", a descriptor must have at least one bit set in each field.) In general, $S_1 \cup S_2$ will not be convex, and similarly $D_1 \vee D_2$ need not be a convex descriptor. Moreover $D_1 \vee D_2$ will not, in general, denote $S_1 \cup S_2$. If $D_1 \vee D_2$ is convex, it denotes the *convex hull* of $S_1 \cup S_2$.

In Fig. 1, S_1 is the convex hull of the cells a:(11, 13) and b:(12, 12). S_2 is the convex hull of the points c:(5, 8), d:(6, 7), e:(6, 6), f:(7, 6) and g:(5, 5). See Fig. 1(c). These are convex clusters. By the *content* of a cluster, we will mean the number of items in it. For example, regarding S_2 as a cluster of the items c, d, e, f and g implies that content $(S_2) = 5$.

We will assume the following *clustering algorithm*. Let a new item I be entered into the space at a cell with descriptor D_I. Let D_C be the descriptor of a cluster C.

Condition 1. The item I will be added to C only if $D_I \vee D_C$ is convex. $D_I \vee D_C$ will be the new cluster descriptor.

Condition 2. If there are two, or more, clusters C_1 and C_2 such that $D_I \vee D_{C_1}$ and $D_I \vee D_{C_2}$ are both convex, then I will be added to a cluster of minimal content, i.e., containing the fewest items. This need not be a cluster of minimal extent, i.e., the smallest $b_1 \times b_2 \times \cdots \times b_m$-cube. Note that, whenever an item is added to an existing cluster C, its content is increased by one, its extent may, or may not, increase. Two items may be entered into the same cell of the space, but they need not belong to the same cluster. Condition 2 will not be used until we begin the analysis of § 6.

Finally note that with this algorithm, which dynamically creates clusters as items are entered and does not redefine cluster boundaries after the fact, the nature of the formed clusters is dependent on the order of entry. For example, the items (1, 1), (2, 2), (3, 3), (4, 4) entered in this order form a single cluster with descriptor (11110 $\cdots$ 11110 $\cdots$); but entered in the order (1, 1), (4, 4), (2, 2), (3, 3) form two distinct clusters with descriptors (11000 $\cdots$ 11000 $\cdots$) and (00110 $\cdots$ 00110 $\cdots$).

3. Halos and intersections. In this section, we calculate two values that will be used in the probabilistic arguments of succeeding sections. Let C be any $b_1 \times b_2 \times \cdots \times b_m$ cluster. If a new item is associated with a cell that is contained within, or is adjacent to, C then that item *may* be added to C. The set of cells adjacent to C we call its *halo*. Combined with its halo, the *effective extent* of C is $(b_1+\varepsilon_1) \times (b_2+\varepsilon_2) \times \cdots \times (b_m+\varepsilon_m)$, where $1 \leqq \varepsilon_j \leqq 2$, $1 \leqq j \leqq m$.

For field$_j$, ε_j counts the number of adjacent 0 bits in the descriptor, so for any particular cluster C, $\varepsilon_j = 1$ or 2, depending on whether C is centrally placed in the space with respect to attribute-j. (Note that we will always assume that $b_j \leqq w_j - 1$, since otherwise all bits in field$_j$ are set. Consequently we may assume $\varepsilon_j \neq 0$.) Now we need an expression for $\bar{\varepsilon}_j$, the *expected extent* of a halo, in terms of b_j, the extent of C.

Consider field$_j$ in D_C for any attribute j. If the string of b_j consecutive bits is in the "interior" of the field, there are adjacent unset bit positions at either end, and $\varepsilon_j = 2$. If the string is in either "extreme" position there is only one adjacent unset bit position and $\varepsilon_j = 1$. We may thus let

$$\begin{aligned} \bar{\varepsilon}_j &= 1 \cdot \text{prob (extreme config.)} + 2 \cdot \text{prob (interior config.)} \\ &= 1 \cdot (b_j+1)/w_j + 2 \cdot (w_j - b_j - 1)/w_j \\ &= (2w_j - b_j - 1)/w_j = 2 - (b_j+1)/w_j, \end{aligned} \tag{3.1}$$

where the probabilities of an "extreme" or "interior" configuration can be calculated by examining the generation sequences of the possible configurations. We will assume that (3.1) is valid even when using $\bar{b}_j$, the *expected number* of bits set in field j. Clearly $\bar{\varepsilon}_j$ is a function of $\bar{b}_j$, with $\bar{\varepsilon}_j \to 2$, when $\bar{b}_j \to 1$ and $\bar{\varepsilon}_j \to 1$ when $\bar{b}_j \to w_j - 1$.

Let C_1 and C_2, be two clusters, both of extent $b_1 \times b_2 \times \cdots \times b_m$. We will determine their expected *effective* intersection, that is, the expected number of cells in the intersection of C_1 and C_2 *together with their halos*. (Note that we have considerably simplified the more general problem of expected intersection of clusters by assuming that C_1 and C_2 are of identical extent. But it will suffice for our purposes.)

Again consider field$_j$ in D_{C_1} and D_{C_2}, for any attribute j, $1 \leqq j \leqq m$. Let s_1 and s_2 denote the strings of b_j consecutive 1-bits in each descriptor field respectively. Assume that s_1 is positioned anywhere in field$_j$ and that s_2 is displaced exactly d positions *to the left* of s_1, $0 \leqq d \leqq w_j - b_j$.

For various values of d one can show that the amount of effective overlap (of the descriptors and their halos) of s_1 and s_2 in field$_j$ is

$$\text{overlap } (d)\begin{cases} = b_j + \bar{\varepsilon}_j & \text{if } d = 0, \\ = b_j - d + 2 & \text{if } 1 \leqq d \leqq b_j + 1, \\ = 0 & \text{if } d > b_j + 1. \end{cases}$$

The probability of obtaining a displacement of $d \geqq 0$ bits is the probability of locating s_1 in any position with at least d positions to its left times the probability of locating s_2 in that position which is precisely offset by d bits. Thus

$$\text{prob (offset} = d) = \frac{(w_j - d - b_j + 1)}{(w_j - b_j + 1)^2}$$

hence the expected overlap in field$_j$ expressed in terms of the possible displacements d of s_2 relative to s_1 is

$$\begin{aligned} \text{exp (overlap)} &= \text{overlap } (0) \cdot \text{prob } (0) + \sum_{d=1}^{b_j+1} \text{overlap } (d) \cdot \text{prob } (d) \\ &= (b_j + \varepsilon_j) + \sum_{d=1}^{b_j+1} \left[\frac{(b_j - d + 2)(w_j - d - b_j + 1)}{(w_j - b_j + 1)^2} \right] \end{aligned} \tag{3.2}$$

(which replacing the last term with the sum of the finite series becomes)

$$= b_j + \frac{2w_j - b_j - 1}{w_j} + \frac{(b_j + 1)(b_j + 2)(3w_j - 4b_j)}{3 \cdot (w_j - b_j + 1)^2}.$$

Although we have derived (3.2) under the assumption that b_j is integral, we again assume that the expression holds for real values, $\bar{b}_j$. (One of the most convincing ways of verifying (3.1) and (3.2) is to work out and exhaustively count all possible configurations for small values of b_j and w_j, possibly for $b_j = 2$, $w_j = 7$.)

Let $\bar{\gamma}(n)$ denote the expected number of convex clusters given n items in the space. Assume that n items, chosen independently from a uniform distribution on the space, have been entered into the space, and that an $(n+1)$-st item I is entered. According to the clustering algorithm of the preceding section, item I will be added to an existing cluster if $D_I \vee D_C$ is convex for some cluster C. Thus item I will create a new singleton cluster if and only if $D_I \vee D_C$ is not convex for all C, and so one has the difference equation

$$\bar{\gamma}(n+1) = \bar{\gamma}(n) + \Delta\gamma(n), \tag{3.3}$$

where $\Delta\gamma(n) = \text{prob } (\forall C, D_I \vee D_C \text{ is not convex})$, and $\bar{\gamma}(1) = 1$.

Since $D_I \vee D_C$ is convex if and only if each field j is convex

$$\text{prob } (D_I \vee D_C \text{ is convex}) = \prod_{j=1}^{m} \frac{(\bar{b}_j + \bar{\varepsilon}_j)}{w_j} \tag{3.4}$$

so that

$$\begin{aligned} \rho(n) &= \text{prob } (D_I \vee D_C \text{ is not convex}) \\ &= 1.0 - \prod_{j=1}^{m} \frac{(\bar{b}_j + \bar{\varepsilon}_j)}{w_j} \end{aligned} \tag{3.5}$$

and thus

$$\begin{aligned}\Delta\gamma(n) &= \text{prob}\,(\forall C, D_I \vee D_C \text{ is not convex})\\ &= (\text{prob}\,(D_I \vee D_C \text{ is not convex}))^{\bar{\gamma}(n)}\\ &= \rho(n)^{\bar{\gamma}(n)} = \left(1.0 - \prod_{j=1}^{m} \frac{(\bar{b}_j + \bar{\varepsilon}_j)}{w_j}\right)^{\bar{\gamma}(n)}.\end{aligned} \tag{3.6}$$

Note that expressions (3.4), (3.5) and (3.6) are valid only if the distributions of bits set in distinct fields are uniform and independent. Initially we can assume this uniformity. But in § 6 we will have to recalculate (3.5) under a condition in which the clusters are no longer uniformly distributed throughout the space. Independence will be assumed throughout this paper, even though it is well known that in any particular application the various attributes of the items represented in the space may exhibit considerable dependency. Note also, that $\bar{\gamma}(n)$ and $\rho(n)$ are functions of n, the number of items entered into the space as we have clearly indicated. So also are $\bar{b}_j$ and $\bar{\varepsilon}_j$; but to keep the expressions to a reasonable size we have not always used either $\bar{b}_j(n)$ or $\bar{\varepsilon}_j(n)$.

The difference equation (3.3) can now be simply expressed as

$$\bar{\gamma}(n+1) = \bar{\gamma}(n) + \rho(n)^{\bar{\gamma}(n)}. \tag{3.7}$$

4. Expected number of convex clusters. If we have an expression for $\bar{b}_j$ as a function of n, then we can calculate $\bar{\varepsilon}_j$ using (3.1), $\rho(n)$ using (3.5), and $\bar{\gamma}(n)$, the expected number of convex clusters, using (3.7). We will set up and solve a difference equation of the form

$$\bar{b}_j(n+1) = \bar{b}_j(n) + \Delta\bar{b}_j(n). \tag{4.1}$$

However, we will have to obtain $\Delta\bar{b}_j$ in a somewhat indirect manner.

Let $\bar{B}_j$ denote the expected *total* number of 1-bits set in field j of *all* cluster descriptors D_C. Then

$$\bar{b}_j(n) = \bar{B}_j(n)/\gamma(n) \quad \text{and} \quad \Delta\bar{b}_j(n) = (\gamma(n)\cdot\Delta\bar{B}_j(n) - \bar{B}_j(n)\cdot\Delta\gamma(n))/\gamma(n)^2. \tag{4.2}$$

We may approximate $\gamma(n)$ by $\bar{\gamma}(n)$ and using (3.6), $\Delta\gamma(n)$ by $\rho(n)^{\bar{\gamma}(n)}$. We thus need only an expression for $\Delta\bar{B}_j$.

Again we look at the incremental change in B_j resulting from the entry of a new item, I. If for all clusters C, D_I, D_C is not convex, then I will start a new cluster, and B_j will be increased by 1. If for some C, D_I, D_C is convex, then a new bit may, or may not, be set in the resulting descriptor field depending on the position of the bit in the field. The probability of adding an extra bit and incrementing B_j is

$$\begin{aligned}\text{prob}\,(\text{incrementing } B_j \,|\, D_I \vee D_C \text{ is convex}) &= \varepsilon_j/(b_j+\varepsilon_j)\\ &= (2w_j - b_j - 1)/[(b_j+2)w_j - b_j - 1].\end{aligned}$$

(Note that evaluating this expression for $b_j = w_j - 1$ and $b_j = 1$ provides the bounds $1/w_j \leqq \text{prob} < \frac{2}{3}$.) Thus the incremental change to $\bar{B}_j$ as the result of entering a new item is given by

$$\begin{aligned}\Delta\bar{B}_j(n) = 1 \cdot &\text{prob}\,(VC, D_I \vee D_C \text{ is not convex})\\ &+ \text{prob}\,(\text{incrementing } B_j)\cdot\text{prob}\,(\exists C, D_I \vee D_C \text{ is convex})\end{aligned} \tag{4.3}$$

or

$$\Delta\bar{B}_j(n) = \rho^{\bar{\gamma}} + \left[\frac{2w_j - \bar{b}_j - 1}{(\bar{b}_j + 2)w_j - \bar{b}_j - 1}\right] \cdot (1 - \rho^{\bar{\gamma}}). \tag{4.4}$$

(Note that in practice, (4.4) may be an upper bound for $\Delta\bar{B}_j$, since for large n we expect there may be several clusters C to which the item I could be added, and a reasonable entry procedure might choose that "best" cluster C which minimizes the increase in cluster extent, and thus $\bar{B}_j$.)

Now, using (4.2)

$$\begin{aligned}\Delta\bar{b}_j(n) &= (\bar{\gamma}(n) \cdot \Delta\bar{B}_j(n) - \bar{B}_j(n) \cdot \Delta\bar{\gamma}(n))/\bar{\gamma}(n)^2 \\ &= \frac{(\bar{b}_j^2 + \bar{b}_j)\rho^{\bar{\gamma}} - ((\bar{b}_j^2 + \bar{b}_j)\rho^{\bar{\gamma}} - 2)w - \bar{b}_j - 1}{((\bar{b}_j + 2)w_j - \bar{b}_j - 1)\bar{\gamma}}.\end{aligned} \tag{4.5}$$

Expressions (3.7), (4.1) and (4.5) may now be used to set up a set of $m+1$ difference equations in $(\bar{\gamma}, \bar{b}_1, \cdots, \bar{b}_m)$ which, since the independent variable is an integer, may be just evaluated iteratively. These are:

$$\begin{aligned}&\bar{\gamma}(n+1) = \bar{\gamma}(n) + \Delta\gamma(n) = \bar{\gamma}(n) + \rho(n)^{\bar{\gamma}(n)}, \\ &\bar{b}_j(n+1) = \bar{b}_j(n) + \Delta\bar{b}_j(n) \\ &\quad = \bar{b}_j(n) + \frac{(\bar{b}_j(n)^2 + \bar{b}_j(n))\rho(n)^{\bar{\gamma}(n)} - [(\bar{b}_j(n)^2 + \bar{b}_j(n))\rho(n)^{\bar{\gamma}(n)} - 2]w - \bar{b}_j(n) - 1}{((\bar{b}_j^2 + 2)w - \bar{b}_j - 1)\bar{\gamma}(n)},\end{aligned} \tag{4.6}$$

where $\bar{\gamma}(1) = 1$, $\bar{b}_1(1) = \cdots = \bar{b}_m(1) = 1$, and using (3.5)

$$\rho(n) = 1.0 - \prod_{j=1}^{m}\left[\frac{(\bar{b}_j(n) + 2)w_j - \bar{b}_j(n) - 1}{w_j^2}\right].$$

Tables 2 (a, b, c) illustrate numerical solutions for three sets of such difference equations. Initially nearly every new item begins a new cluster. The probability that no $D_I \vee D_C$ is convex, that is $\rho^{\bar{\gamma}}$, is close to 1.0. But as $\bar{\gamma}$ and $\Pi_j\bar{b}_j$ (the expected cluster extent) increase, the likelihood of an item I being added to an existing cluster increases.

TABLE 2(a)
Numbers and extents of unbounded clusters in the space (8, 6, 10, 8).

	w_j	8	6	10	8
n	$\bar{\gamma}(n)$	$\bar{b}_1(n)$	$\bar{b}_2(n)$	$\bar{b}_3(n)$	$\bar{b}_4(n)$
10	9.3	1.047	1.046	1.048	1.047
20	17.4	1.095	1.094	1.096	1.095
40	30.5	1.193	1.190	1.195	1.193
60	40.4	1.292	1.286	1.295	1.292
100	53.5	1.489	1.488	1.505	1.489
150	61.8	1.772	1.753	1.783	1.772
200	65.4	2.056	2.027	2.073	2.056
300	67.3	2.617	2.564	2.647	2.617
400	67.5	3.315	3.045	3.171	3.125
500	67.5	3.573	3.465	3.636	3.573

$m = 4$, $\Pi_j w_j = 3.840$

TABLE 2(b)
Numbers and extents of unbounded clusters in the space (4, 7, 10, 15, 20).

	w_j	4	7	10	15	20
n	$\bar{\gamma}(n)$	$\bar{b}_1(n)$	$\bar{b}_2(n)$	$\bar{b}_3(n)$	$\bar{b}_4(n)$	$\bar{b}_5(n)$
10	9.9	1.006	1.006	1.006	1.006	1.006
20	19.6	1.011	1.012	1.012	1.012	1.012
40	38.6	1.023	1.024	1.024	1.025	1.025
60	56.8	1.034	1.036	1.037	1.037	1.037
100	91.2	1.057	1.061	1.062	1.062	1.063
150	130.7	1.086	1.092	1.093	1.095	1.096
200	166.6	1.117	1.123	1.126	1.127	1.128
300	228.7	1.178	1.188	1.192	1.195	1.196
400	279.6	1.241	1.255	1.261	1.265	1.267
500	320.9	1.307	1.326	1.333	1.338	1.340
600	354.1	1.375	1.399	1.408	1.415	1.418
700	380.4	1.445	1.475	1.486	1.494	1.498
800	401.0	1.518	1.554	1.567	1.576	1.581
900	416.8	1.593	1.635	1.650	1.662	1.667
1,000	428.7	1.669	1.719	1.736	1.750	1.756
1,500	453.0	2.062	2.152	2.185	2.209	2.222
2,000	456.1	2.431	2.571	2.623	2.661	2.679

$m = 5$, $\Pi_j w_j = 84{,}000$

TABLE 2(c)
Numbers and extents of unbounded clusters in the space (5, 10, 15, 20, 25, 30).

	w_j	5	10	15	20	25	30
n	$\bar{\gamma}(n)$	$\bar{b}_1(n)$	$\bar{b}_2(n)$	$\bar{b}_3(n)$	$\bar{b}_4(n)$	$\bar{b}_5(n)$	$\bar{b}_6(n)$
500	494.3	1.007	1.007	1.007	1.008	1.008	1.008
1,000	977.2	1.014	1.015	1.015	1.015	1.015	1.015
1,500	1,449.1	1.021	1.022	1.023	1.023	1.023	1.023
2,000	1,909.9	1.029	1.030	1.031	1.031	1.031	1.031
3,000	2,799.6	1.044	1.046	1.046	1.046	1.047	1.047
4,000	3,647.5	1.059	1.061	1.062	1.062	1.063	1.063
5,000	4,455.2	1.074	1.077	1.078	1.079	1.079	1.079
10,000	7,934.0	1.154	1.161	1.163	1.164	1.165	1.166
15,000	10,585.1	1.241	1.253	1.257	1.258	1.259	1.260
20,000	12,538.7	1.335	1.353	1.359	1.361	1.363	1.364
30,000	14,827.8	1.550	1.582	1.592	1.596	1.599	1.601
40,000	15,705.5	1.791	1.841	1.857	1.864	1.869	1.872
50,000	15,936.6	2.040	2.119	2.135	2.146	2.152	2.157
60,000	15,975.1	2.279	2.377	2.407	2.421	2.430	2.435
70,000	15,978.9	2.502	2.625	2.663	2.681	2.692	2.700
80,000	15,979.1	2.707	2.858	2.904	2.927	2.940	2.949
90,000	15,979.2	2.900	3.077	3.132	3.159	3.175	3.186
100,000	15,979.2	3.079	3.285	3.349	3.380	3.400	3.411

$m = 6$, $\Pi_j w_j = 11{,}250{,}000$

Indeed, with the configuration of Table 2(a), which represents a fairly small 4-dimensional space of $\Pi_j w_j = 8 \cdot 6 \cdot 10 \cdot 8 = 3{,}840$ points, by the time $n = 300$ items have been entered the entire space has been effectively covered by 67 existing clusters. $\rho^{\bar{\gamma}} \to 0.0$ and $\bar{\gamma}(n) \to$ constant, since all newly entered items are included in an existing cluster. But $\bar{b}_j(n)$ keeps increasing as the extent of individual clusters keeps growing.

Table 2(b) illustrates the behavior of clusters in a slightly larger 5-dimensional space of $\Pi_j w_j = 4 \cdot 7 \cdot 10 \cdot 15 \cdot 20 = 84{,}000$ cells. It is much the same, except that when $n = 2{,}000$ an expected $\bar{\gamma} = 456$ clusters have been formed with an expected content of $n/\bar{\gamma} = 4.38$ items and expected extent of $\Pi_j \bar{b}_j = 116{,}207$ cells per cluster. At $n =$ 1,500 there were virtually the same number of clusters, with expected content $=3.31$ but with an expected extent of only 47.15.

In a still larger 6-dimensional space of $\Pi_j w_j = 5 \cdot 10 \cdot 15 \cdot 20 \cdot 25 \cdot 30 =$ 11,250,000 cells, after $n = 50{,}000$ items have been entered there will be approximately $\bar{\gamma} = 15{,}936$ distinct clusters with an expected content of only 3.14 items per cluster. After 100,000 items have been entered there will be nearly the same number of clusters, but now each cluster will contain an expected 6.26 items.

5. Practical implications. Bit descriptors, as developed in this paper, can be the key to very efficient multi-attribute file retrieval methods. We are primarily concerned with a retrieval method called "indexed descriptor access" described in detail in [8]. References [1, 7, 10, 11] represent a sample of other multi-attribute retrieval methods which are similar, or related in some way. Virtually all retrieval methods work most effectively when "similar" items are stored together; that is are clustered in some fashion. The item (or record) entry algorithm of § 2 is one way of dynamically organizing any data file.

The solution of the system of $m+1$ difference equations in the preceding § 4, that calculate $\bar{\gamma}$ and $\bar{b}_j$, for $1 \leqq j \leqq m$, provide precisely the information needed to estimate expected retrieval costs, assuming that we will let clusters contain an arbitrary number of items. In [8] it is shown that

$$\bar{\gamma}(n) \cdot \prod_{j \in Q} \frac{\bar{b}_j}{w_j} \tag{5.1}$$

denotes the expected number of accesses in the data file to respond to a query Q which conjunctively specifies some, or all, of the m item attributes. (Here, $\Pi_{j \in Q}$ denotes a product involving those fields j specified in the query.)

If, for example, all m attributes were specified in a retrieval query (normally denoting at most a single record of the data file), then using the final values in Tables 2(a, b, c) and (5.1), the expected storage accesses to the data file are:

file 1	(500 items)	2.831 accesses,
file 2	(2,000 items)	0.631 accesses,
file 3	(100,000 items)	1.886 accesses.

Note that these values do not denote the total cost of retrieval using indexed descriptor access. A few additional accesses must be made in one, or more, index files. But it is not the purpose of this paper to discuss any retrieval method in detail. Still we do want to indicate (1) that this dynamic clustering procedure can be used in conjunction with an effective retrieval scheme; (2) that the resulting analyses can be used to estimate expected retrieval costs; and (3) that retrieval in very large files need

be no more expensive than much smaller files—provided they are represented over a sufficiently large attribute space.

The expressions of the preceding section provide the means for numerically calculating $\bar{\gamma}(n)$ and $\bar{b}_j(n)$ as in Tables 2(a, b, c) and thereby calculating the average number of items per cluster, $n/\bar{\gamma}(n)$. But the tables do not indicate the extreme variability in either the extent or the content of these clusters. A few clusters have large content, while most contain only one or two items. Intuitively, of the many initially scattered singleton clusters, chance dictates that a few will begin to grow. But then the probability that new items will be contained in, or adjacent to, these larger clusters (that is, $D_I \vee D_C$ will be convex) is greater, and so they tend to keep growing; at least until the space is well covered by clusters.

If we are modeling an actual computer file organization in which items (or records) with similar attributes are to be clustered (blocked) together in a data file, then the presence of large clusters is unrealistic. There will be some finite upper limit on the number of records that can form a physical block of the data file. Thus we really want to derive the results of the preceding section, subject to the constraint that "no cluster can contain more than kmax items".

In the next section we add this bounding constraint, and in so doing must use a different analytic approach. The expected number of clusters with precisely k items, $1 \leqq k \leqq \text{kmax}$, must be determined. The results are somewhat surprising, and in many respects more revealing of the behavior of this clustering process in actual practice.

6. Number of clusters containing precisely k items. A *k-cluster*, C_k, is one containing precisely k items, $1 \leqq k \leqq \text{kmax}$. kmax denotes the maximum possible content of (number of items in) any cluster. To express the expected number of these k-clusters and their extents we could use a notation such as $\bar{\gamma}^{(k)}(n)$, $\bar{b}_j^{(k)}(n)$ and $\bar{\varepsilon}_j^{(k)}(n)$ to indicate a functional dependence on both k and n. But rigorous adherence to this notation would lead to unwieldy expressions below. Moreover, we will show that $\bar{b}_j^{(k)}$ and $\bar{\varepsilon}_j^{(k)}$ are not, in fact, dependent on n. Consequently, we will use an abbreviated notation $\bar{\gamma}(k)$, $\bar{b}_j(k)$ and $\bar{\varepsilon}_j(k)$ and let the dependence on n in the case of $\bar{\gamma}(k)$ be tacitly understood. Now

$$\bar{\gamma} = \sum_{k=1}^{\text{kmax}} \bar{\gamma}(k), \qquad \bar{B}_j = \sum_{k=1}^{\text{kmax}} \bar{b}_j(k) \cdot \bar{\gamma}(k), \qquad \bar{b}_j = \frac{\bar{B}_j}{\bar{\gamma}}, \tag{6.1}$$

where the terms $\bar{\gamma}$, $\bar{B}_j$ and $\bar{b}_j$ on the left are identical to $\bar{\gamma}(n)$, $\bar{B}_j(n)$ and $\bar{b}_j(n)$ of preceding sections.

We now concentrate on the behavior of just k-clusters. By definition, for all 1-clusters, $\bar{b}_j(1) = 1.0$, for $1 \leqq j \leqq m$. A new 2-cluster is formed by adding an item to an existing 1-cluster. For each field j there is a distinct probability that the entered item will add another bit to that field of the cluster descriptor. And since all 2-clusters are formed in the same way, the expected number of bits per field in any single 2-cluster descriptor is that of all 2-clusters as a whole. Thus

$$\bar{b}_j(2) = \bar{b}_j(1) + \left[1 - \frac{\bar{b}_j(1)}{\bar{b}_j(1) + \bar{\varepsilon}_j(1)}\right],$$

where the expression in brackets denotes the probability of adding a new bit to field$_j$ of the cluster descriptor, assuming $D_I \vee D_C$ is convex; and where $\bar{\varepsilon}_j(1)$ is calculated from $\bar{b}_j(1)$ using (3.1). In general, for $2 \leqq k \leqq \text{kmax}$

$$\bar{b}_j(k) = \bar{b}_j(k-1) + [1.0 - \bar{b}_j(k-1)/(\bar{b}_j(k-1) + \bar{\varepsilon}_j(k-1))]. \tag{6.2}$$

Note that $\bar{b}_j(k)$, the expected number of bits set in field$_j$ of a k-cluster, and $\bar{\varepsilon}_j(k)$ are both independent of n or $\gamma(k)$, the number of such clusters.

Let I be a new item entered into the space. Following the clustering algorithm of § 2, by Condition (1) I will be added to some existing C_k only if $D_I \vee D_{C_k}$ is convex, and by Condition (2) if $D_I \vee D_{C_j}$ is *not* convex for any $1 \leqq j < k$. The item I will start a new singleton cluster only if $D_I \vee D_{C_k}$ is not convex for any C_k, $1 \leqq k \leqq \text{kmax}-1$. (Note that I cannot be added to any kmax-cluster.) Consequently,

$$\begin{aligned}&\text{prob (adding } I \text{ to a } k\text{-cluster, } 1 \leqq k < \text{kmax})\\ &\quad = \text{prob } (\forall C_j, I < k, D_I \vee D_{C_I} \text{ is not convex}) \cdot \text{prob } (\exists C_k, D_I \vee D_{C_k} \text{ is convex}).\end{aligned} \tag{6.3}$$

And, in general, the incremental change in $\bar{\gamma}(k)$ as a result of entering a new item I into the space can be expressed by

$$\Delta\bar{\gamma}(k) = \text{prob (adding } I \text{ to a } (k-1)\text{-cluster} - \text{prob (adding } I \text{ to a } k\text{-cluster}). \tag{6.4}$$

The second term accounts for the fact that every time a $(k+1)$-cluster is formed from a k-cluster, $\gamma(k)$ is decreased by one. This expression must be modified in the case of $k = 1$ and $k = \text{kmax}$. When $k = 1$, the first term of (6.4) is the probability that $D_I \vee D_{C_i}$ is not convex *for any cluster*, $i < \text{kmax}$. When $k = \text{kmax}$, the second term is simply omitted.

The probabilities of (6.3) may be calculated in two different ways. First we assume, as in preceding sections, that k-clusters are uniformly distributed through the space in an independent manner, so that using (3.4) we have

$$\text{prob } (D_I \vee D_{C_k} \text{ is convex}) = \prod_{j=1}^{m} (\bar{b}_j(k) + \bar{\varepsilon}_j(k))/w_j,$$

where again this probability is independent of n. As in §§ 3 and 4 we let $\rho(k)$ denote the probability that $D_I \vee D_{C_k}$ is not convex for any particular k-cluster. Thus (6.3) becomes

$$\text{prob (adding } I \text{ to a } k\text{-cluster, } 1 \leqq k < \text{kmax}) = \prod_{i<k} \rho(j)^{\bar{\gamma}(j)} \cdot [1.0 - \rho(k)^{\bar{\gamma}(k)}], \tag{6.5}$$

which when substituted into (6.4) yields the following difference expressions

$$\Delta\bar{\gamma}(1) = \prod_{k<\text{kmax}} \rho(k)^{\bar{\gamma}(k)} - (1.0 - \rho(1)^{\bar{\gamma}(1)})$$

$$\begin{aligned}\Delta\bar{\gamma}(k) &= \prod_{i<k-1} \rho(j)^{\bar{\gamma}(j)} \cdot (1.0 - \rho(k-1)^{\bar{\gamma}(k-1)}) - \prod_{i<k} \rho(j)^{\bar{\gamma}(j)} \cdot (1.0 - \rho(k)^{\bar{\gamma}(k)})\\ &= \prod_{i<k-1} \rho(j)^{\bar{\gamma}(j)} \cdot [(1.0 - \rho(k-1)^{\bar{\gamma}(k-1)}) - \rho(k-1)^{\bar{\gamma}(k-1)} \cdot (1.0 - \rho(k)^{\bar{\gamma}(k)})]\end{aligned}$$

for $2 \leqq k \leqq \text{kmax}-1$, and

$$\Delta\bar{\gamma}(\text{kmax}) = \prod_{i<\text{kmax}-1} \rho(j)^{\bar{\gamma}(j)} \cdot [1.0 - \rho(\text{kmax}-1)^{\bar{\gamma}(\text{kmax}-1)}]. \tag{6.6}$$

With these one can set up kmax difference equations of the form

$$\bar{\gamma}^{(k)}(n+1) = \bar{\gamma}^{(k)}(n) + \Delta\bar{\gamma}(k) \tag{6.7}$$

and evaluate them iteratively using $\bar{\gamma}^{(1)}(1) = 1.0$, $\bar{\gamma}^{(k)}(1) = 0.0$, $2 \leqq k \leqq \text{kmax}$, as initial conditions.

The iterative evaluation of this set of equations yields values which are reasonably close to those values obtained in practice. For example, the maximum relative error

between the expected $\bar{\gamma}(n)$ calculated in this fashion, and the actual $\gamma(n)$ observed in more than 50 generated clustered files over different attribute spaces is less than 0.19.

However, it is intuitively evident that k-clusters may not be independently distributed throughout the space. This leads to a second way of determining at least the first two differences of (6.6).

Consider just the case of 1-clusters. A new 1-cluster will be created by a newly entered item only if it is not within or adjacent to any existing k-cluster, $k < \text{kmax}$. 1-clusters are *formed* only in that portion of the space not covered by k-clusters. (Note that in this dynamic process, 1) 1-clusters which were formed at any earlier time may *exist* in other portions of the space if they have not been subsequently enlarged; and 2) once a cluster has its maximal possible content, kmax, it no longer affects the clustering process—it is effectively no longer there.)

Now for 1-clusters we can re-evaluate (6.3) as

$$\begin{aligned}\text{prob (adding } I \text{ to a 1-cluster)} &= \text{prob } (\exists \text{ a 1-cluster } C, D_I \vee D_C \text{ is convex}) \\ &= \sum_j \text{prob } (D_I \vee D_{C_j} \text{ is convex}) \\ &\quad - \sum_{i,j} \text{prob } (D_I \vee D_{C_i} \text{ and } D_I \vee D_{C_j} \text{ are convex}) \\ &\quad + \sum_{i,j,k} \cdots,\end{aligned}$$

where the sums are run over all 1-clusters C_i, C_j, C_k. The first term of this expansion is simply

$$\bar{\gamma}(1) \cdot \prod_j (\bar{b}_j(1) + \bar{\varepsilon}_j(1)) \Big/ \prod_j w_j.$$

If the 1-clusters were uniformly distributed then the second term would be

$$\binom{\bar{\gamma}(1)}{2} \cdot \prod_j \exp \text{ (overlap in field}_j) \Big/ \prod_j w_j$$

into which we could substitute (3.2).

But (3.2) was derived assuming that the clusters were distributed over the entire space. They are not. They are confined to a much smaller portion of the space, and the expected intersection is higher than that predicted by (3.2). We can approximately account for this by replacing the actual field widths, w_j, in (3.2) with apparent field widths, call them w'_j, where $\prod_j w'_j =$ total volume of space *not* covered by k-clusters, $2 \leqq k \leqq \text{kmax}$. We approximate this reduction factor by

$$F_1 = \prod_{k \geqq 2} \rho(k)^{\bar{\gamma}(k)}.$$

Then we set $w'_j = F_1^{1/m} \cdot w_j$. Since $\bar{b}_j(1) = 1.0$ for all j, (3.2) becomes, in the case of 1-clusters

$$\exp \text{ (overlap)} = \frac{1}{w_j'^2} [3w_j'^2 + 4w'_j - 8]$$

and (6.3) becomes

$$\begin{aligned}&\text{prob (adding } I \text{ to a 1-cluster)} \\ &= \frac{1}{\prod_j w_j} \cdot \left[\bar{\gamma}(1) \cdot \prod_j (\bar{b}_j(1) + \bar{\varepsilon}_j(1)) - \binom{\bar{\gamma}(1)}{2} \cdot \prod_j (3w_j'^2 + 4w'_j - 8/w_j'^2 + \cdots)\right],\end{aligned} \tag{6.8}$$

where the third and succeeding terms of the expansion are ignored.

There are enough approximations in this refined derivation to make it unreliable for larger k-clusters. But observation of the clustering process indicates that the distribution of k-clusters becomes more nearly uniform as $k \to \text{kmax}$ anyway. We have found that use of the alternate computation of prob (adding I to a 1-cluster) in the first two differences of (6.6) leads to slightly more accurate expected values when compared to actual behavior. This refinement was used to generate Figs. 3(a, b, c) and Tables 5 and 6 of the following section.

However calculated, the behavior of the solutions of the set of difference equations (6.7) is quite interesting. As one would intuitively expect, only 1-clusters are formed initially. Later, as these begin forming 2-clusters with successively entered items (which in turn serve as the nuclei of 3-clusters), the number of 1-clusters, $\bar{\gamma}(1)$, decreases. As shown in Fig. 3(a) (which denotes the same space as Table 2(a), but with kmax = 5), $\bar{\gamma}(1), \bar{\gamma}(2), \cdots, \bar{\gamma}(\text{kmax}-1)$ all oscillate periodically. But this periodic behavior dies out and an equilibrium is achieved in which $\bar{\gamma}(1), \bar{\gamma}(2), \cdots, \bar{\gamma}(\text{kmax}-1)$ become effectively constant. Only $\bar{\gamma}(\text{kmax})$ will be monotone increasing.

Once equilibrium is attained, the expected number of all clusters, $\bar{\gamma}(n)$, can be expressed by the simple linear equation

$$\bar{\gamma}(n) = c_1 + c_2(n - c_3), \tag{6.9}$$

where $c_1 = \sum_{k<\text{kmax}} \bar{\gamma}(k)$, $c_2 = 1/\text{kmax}$ and $c_3 = \sum_{k<\text{kmax}} k \cdot \bar{\gamma}(k)$.

In Figs. 3(b, c), which duplicate the spaces of Tables 2(b, c) subject to the constraints of kmax = 4 and 3 respectively, the oscillation of $\bar{\gamma}(1), \cdots, \bar{\gamma}(\text{kmax}-1)$ is less apparent, since the amplitude is small and with much longer period; but the steady state behavior is evident.

7. How accurate are these solutions? Given the dimensions $(i_1, i_2, \cdots, i_m)$ of any particular attribute space, one may use (4.6), or (6.7) to derive an expected $\bar{\gamma}(n)$. To test their accuracy a number of files were actually created using the insertion algorithm of § 2 and randomly generated data.

First, we verify the expected number of clusters with unbounded content predicted by (4.6). The expected $\bar{\gamma}(n)$ for spaces of dimensions (8, 6, 10, 8) given in Table 2(a). Table 4 compares those expected values with those of the test files. Five test files of 500 records each were created with no constraint on maximum cluster content, $\bar{\gamma}(n)$ is seen to be consistently greater than $\gamma(n)$. The derivation of (4.6) takes no account of the variability of either cluster extent or content. All probabilities are based on an average cluster content $\prod_j \bar{b}_j$. However, the creation of unbounded clusters by the entry algorithm of § 2 tends to be unstable with respect to perturbations from the mean. A cluster which is already large is more likely to "capture" a newly entered item than a smaller cluster. Thus large clusters tend to become very large, eventually dominating the space, while the remaining clusters remain small. Condition (2) of the entry procedure minimizes this instability somewhat, but not completely.

If it were practical to create unbounded clusters and retrieve items from them, then either a revised algorithm or a refined analysis would be essential. Since this is not so, Table 4 (and similar unprinted comparisons) should be sufficient to convince us of at least the basic correctness of (4.6). In contrast, with bounded cluster creation as considered in the preceding section, the behavior of the entry algorithm is "self-correcting" with respect to cluster extent. Large clusters still grow more rapidly. But once their content reaches kmax, they no longer affect the entry procedure; they cannot dominate the space.

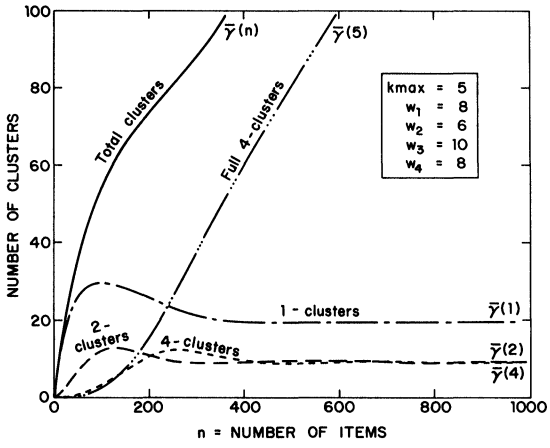

FIG. 3(a)

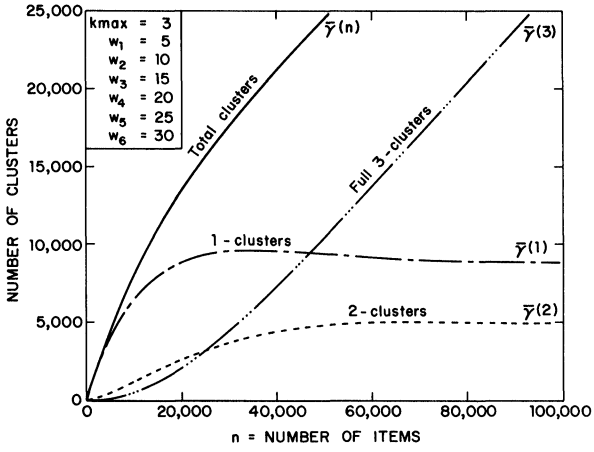

FIG. 3(b)

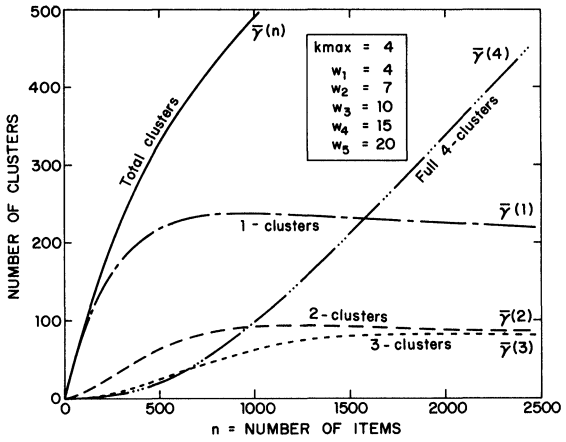

FIG. 3(c)

TABLE 4

Comparison of expected vs. observed numbers of clusters in the space (8, 6, 10, 8), *with unbounded cluster content.*

n	expected $\bar{\gamma}(n)$	observed $\gamma(n)$ file 1	file 2	file 3	file 4	file 5
10	9.4	10	10	9	9	10
20	17.4	17	20	18	18	19
40	24.5	28	31	32	31	31
60	40.4	34	39	39	43	42
80	47.9	37	45	44	46	50
100	53.5	39	51	47	52	53
150	61.8	44	60	54	57	54
200	65.5	46	60	55	60	55
300	67.3	48	63	55	61	59
400	67.5	48	63	55	61	59
500	67.5	48	63	55	61	59

To verify the more important derivation presented in § 6, we have calculated the expected $\bar{\gamma}(n)$ and $\bar{\gamma}(1), \cdots, \bar{\gamma}(\text{kmax})$ using (6.7) for each of the same three spaces given in the preceding examples; and then executed a series of five actual clustering runs for each. Maximum cluster bounds of kmax = 5, 4 and 3 respectively were used. The choice of these bounds is derived from tables 2(a, b, c). These represent, in each case, the average content of a cluster when the space was first "effectively covered". Larger, or smaller, cluster bounds may be dictated by other constraints in individual applications; but in the absence of such constraints these bounds will yield optimal retrieval performance. Indeed, the major value in practice of the derivation of § 4 is to be able to establish approximately optimal cluster bounds.

Table 5 compares the expected numbers of clusters with the observed means of the five test files generated over the space with dimensions (8, 6, 10, 8). Only the expected total clusters, $\bar{\gamma}(n)$, and the expected 1-clusters and 5-clusters, $\bar{\gamma}(1)$ and $\bar{\gamma}(\text{kmax})$ are displayed. The value of $\bar{\gamma}(n)$ is quite accurate. The expected value deviates

TABLE 5

Comparison of expected vs. observed numbers of bounded clusters in the space (8, 6, 10, 8); *max. cluster content* = kmax = 5.

n	expected $\bar{\gamma}(n)$	observed mean	expected $\bar{\gamma}(1)$	observed mean	expected $\bar{\gamma}(5)$	observed mean
20	17.4	17.4	15.2	16.2	0.0	0.0
60	40.3	38.0	27.4	25.4	0.6	0.6
100	54.1	50.2	29.5	28.0	2.8	3.4
200	73.5	69.6	25.7	21.8	15.6	16.6
300	88.1	90.4	21.2	20.0	37.6	37.8
400	106.0	113.2	19.5	16.8	60.2	55.2
500	126.1	127.6	19.6	17.8	80.6	76.2
600	146.2	148.4	19.5	20.0	100.4	98.2
700	166.1	168.4	19.4	21.9	120.4	119.8
800	186.1	188.4	19.5	20.8	140.5	138.8
900	206.1	208.6	19.5	21.6	160.5	156.4
1,000	226.1	229.4	19.5	18.0	180.5	177.6

from the observed mean by at most −7.2 when $n = 400$. This corresponds to a maximum relative error of 0.068.

The estimation of expected k-clusters is somewhat less accurate. An initial periodic behavior as shown in Fig. 3(a) can be seen in all individual runs; but the phase and amplitude of these periods, being largely determined by random events, can be very variable. For example, when $n = 800$, the observed values of $\gamma(1)$ were 21, 17, 17, 17 and 32 with a mean of 20.8, but standard deviation of 5.81. However, by $n = 500$ the predicted equilibrium and consequent general stability are apparent in most individual cluster systems.

Table 6 displays a similar comparison of expected versus observed numbers of clusters in files clustered over the slightly larger space with dimensions (4, 7, 10, 15, 20). The prediction of expected total clusters is somewhat better. A maximum relative deviation from the observed mean of 0.016 occurs when $n = 800$. Again the estimations of $\bar{\gamma}(1)$ and $\bar{\gamma}(4)$ are less accurate.

TABLE 6

Comparison of expected vs. observed numbers of bounded clusters in the space (4, 7, 10, 15, 20); *max. cluster content* = kmax = 4.

n	expected $\bar{\gamma}(n)$	observed mean	expected $\bar{\gamma}(1)$	observed mean	expected $\bar{\gamma}(4)$	observed mean
100	91.0	91.6	83.3	83.6	0.2	0.0
200	165.9	166.4	139.4	139.8	1.7	1.6
300	227.9	229.0	176.6	177.4	5.5	6.0
400	280.2	277.5	200.9	198.8	11.9	12.2
500	325.2	322.2	216.6	211.5	20.9	20.4
600	364.7	365.0	226.5	229.0	32.2	33.4
700	400.0	397.3	232.3	233.0	45.8	45.4
800	432.1	425.2	235.6	226.0	61.4	68.2
900	461.8	455.0	237.1	227.6	78.9	79.0
1,000	489.7	485.2	237.4	229.8	98.0	95.5
1,100	516.2	510.8	236.9	229.3	118.6	119.5
1,200	541.6	537.7	236.0	232.0	140.4	139.8
1,300	566.6	560.6	234.7	235.5	163.3	161.0
1,400	590.6	588.8	233.3	229.0	187.0	179.5
1,500	614.5	614.0	231.8	232.6	211.4	203.0
1,600	638.2	638.9	230.2	234.2	236.3	228.7
1,700	661.8	664.2	228.7	235.4	261.5	249.7
1,800	685.5	687.6	227.3	231.3	287.0	269.8
1,900	709.1	712.0	225.8	234.3	312.6	303.7
2,000	732.9	738.6	224.6	229.7	338.2	326.8

Table 7 was compiled to test the validity of the difference equations (6.7) in large files, where the payoff of any clustering procedure is most pronounced. To generate Table 7, five files of 40,000 items each were clustered over the still larger space of dimensions (5, 10, 15, 20, 25, 30). This represents a space of more than 11 million virtual cells. The maximum relative deviation of the predicted mean $\bar{\gamma}(n)$ from the observed mean is 0.01. That any system of 1-point boundary difference equations should attain this measure of accuracy after 40,000 steps attests to the inherent stability of the difference expressions of (6.6).

TABLE 7

Comparison of expected vs. observed number of bounded clusters in the space (5, 10, 15, 20, 25, 30); *max. cluster content* = kmax = 3.

n	expected $\bar{\gamma}(n)$	observed mean	expected $\bar{\gamma}(1)$	observed mean	expected $\bar{\gamma}(3)$	observed mean
1,000	977.0	987.3	955.0	975.0	0.9	0.8
2,000	1,908.5	1,907.5	1,824.3	1,819.2	7.2	4.7
3,000	2,796.2	2,811.6	2,614.5	2,646.0	22.9	24.1
4,000	3,642.1	3,658.4	3,332.2	3,376.3	48.0	56.8
5,000	4,448.9	4,472.8	3,983.7	4,032.7	86.0	97.2
10,000	7,984.7	8,006.3	6,436.4	6,452.0	467.0	452.5
15,000	10,892.1	10,886.3	7,925.5	7,911.1	1,141.3	1,130.4
20,000	13,374.3	13,387.0	8,802.5	8,810.6	2,053.9	2,043.7
25,000	15,565.1	15,663.2	9,285.4	9,436.4	3,155.2	3,122.3
30,000	17,554.1	17,748.0	9,514.3	9,827.8	4,406.0	4,341.9
35,000	19,403.1	19,672.9	9,582.0	9,983.1	5,775.7	5,684.0
40,000	21,156.1	21,542.4	9,551.5	10,133.7	7,239.4	7,121.8

8. Summary and conclusions. The asymptotic behavior of $\bar{\gamma}(n)$, the expected number of clusters in a finite m-dimensional space, depends on assumptions about the clusters themselves. If the clusters are unbounded, as in § 4, then $\bar{b}_j$, and hence cluster extent, are monotone increasing as a function of n until the space is effectively covered by a few clusters so that $\bar{\gamma}(n) \to c$. And in § 6, where a constraint of a maximum cluster content is imposed, $\bar{\gamma}(n) \to c_1 + c_2(n - c_3)$.

In actual computer applications the latter model is more realistic, since practical considerations invariably impose an upper limit on the number of items that may be clustered in a data file. Its linear behavior has profound implications. In very large data bases, a dynamic clustering algorithm which left many blocks of the data file only partially filled with items would impose intolerable storage overhead. The fact that convex clustering procedures attain an equilibrium, with a fixed upper bound on the number of partially filled physical blocks is crucial. Indeed, the overhead of unused storage, as a percent of the total data file, actually decreases as n becomes large.

Expected storage accesses per query can also be derived from this model by first using (6.2) to calculate $\bar{b}_j(k)$ for $1 \leqq k \leqq$ kmax, and then using (6.1) to calculate $\bar{b}_j(n)$ for $1 \leqq j \leqq m$. With the large file of Fig. 3(c), when $n = 100{,}000$ one obtains $\bar{\gamma}(n) =$ 40,793 and $\bar{b}_j(n) = 1.801$, 1.845, 1.860, 1.867, 1.871 and 1.874 respectively for $j = 1, 2, \cdots, 6$. Comparing these values with those of Table 2(c), one finds more clusters (40.793 vs. 15,979) of smaller extent (40.46 vs. 1327.81 cells per cluster). By using these values, one can now show that to retrieve a single item from the file, an expected $\bar{\gamma}(n) \cdot \prod_{j=1}^{6} \bar{b}_j(n)/w_j = 0.1467$ accesses to the data file will be required. Retrieval performance of this order, which is considerably better than that described in § 5 and which has been verified in actual data files, is impressive.

Readily, the analyses of this paper were undertaken to determine the expected behavior of a particular access and retrieval method (described in [8] and [9]), that represents the attributes of data items by bit descriptors to form a descriptor for a block of several items. This is not an original access method. The first known reference to this technique seems to be [7]. Edgar Cagley independently discovered this approach and, moreover, was the first to implement it in a practical system [5]. Our system closely follows Cagley's design; indeed he provided it, although we use a different clustering mechanism.

Many retrieval systems have employed binary descriptors to represent items; but most form the descriptor by superimposed coding which does not separate distinct attributes; [3] and [10] are representative examples (the latter provides an extensive list of further references). Superimposed coding is superior for a different set of retrieval problems, those for which a single attribute of an item may have a set of values; and leads to a different kind of analysis.

Retrieval of items in an m-dimensional cellular bucket space has been analyzed in [1], [4]. Since the access method is different, neither item descriptors nor clustering is employed, these results are not directly transferable, but there are some definite similarities. In particular they show that as the size of the conceptual attribute space increases, the cost of retrieval decreases.

There is an abundance of "clustering algorithms" in the literature; [2] and [6] provide fine surveys. Most operate over continuous m-dimensional spaces assigning centroids to created clusters and computing metric distances between them. Moreover most require that all the items be known at the time of clustering; newly entered items may cause a restructuring of the clusters. A dynamic data base requires a one-pass procedure that can organize its items "on the fly". Ours is unusual, but hardly unique, cf. [11]. Much of its appeal lies in the fact that it is conceptually simple and easy to implement, that its expected behavior can be analyzed and its performance predicted, and that it works well in practice.

REFERENCES

[1] A. V. AHO AND J. D. ULLMAN, *Optimal partial match retrieval when fields are independently specified*, ACM Trans. Database Sys., 4 (1979), pp. 168–179.

[2] M. R. ANDERBERG, *Cluster Analysis for Applications*, Academic Press, New York, 1973.

[3] W. J. BERMAN, ISIS *users manual*, NASA Tech. Memo. 80144, Hampton, VA, March 1980.

[4] W. J. BERMAN AND J. L. PFALTZ, *Multi-dimensional bucket arrays*, DAMACS Tech. Report TR-16-78, Univ. of Virginia, Charlottesville, 1978.

[5] E. M. CAGLEY, et al., *Information management system reference manual*, GSA/FPA/MCL TM-208, October 1976.

[6] B. EVERITT, *Cluster Analysis*, John Wiley, New York, 1974.

[7] W. D. FRAZER, *A proposed system for multiple descriptor data retrieval*, Some Problems in Information Science, M. Kochen, ed., The Scarecrow Press, Grolier, Danbury, CN, 1965.

[8] J. L. PFALTZ, W. J. BERMAN AND E. M. CAGLEY, *Partial-match retrieval using indexed descriptor files*, Comm. ACM, 23 (1980), pp. 522–528.

[9] J. L. PFALTZ, *Efficient multi-attribute retrieval over very large geographical data files*, Proc. AUTO-CARTO IV, Washington, DC, 1979, pp. 54–62.

[10] C. S. ROBERTS, *Partial-match retrieval via the method of superimposed codes*, Proc. IEEE, 67 (1979), pp. 1624–1642.

[11] G. SALTON AND A. WONG, *Generation and search of clustered files*, ACM Trans. Database Sys., 3 (1978), pp. 321–346.