P. A. Franaszek
J. P. Considine

Reduction of Storage Fragmentation On Direct Access Devices

A technique is described for partially reorganizing the contents of disk storage so as to reduce the level of fragmentation. The method entails choosing that fraction of the contents which is estimated to have the greatest impact on the free space distribution, followed by the relocation of these data to more favorable locations, subject to the system integrity requirements.

1. Introduction

Consider a memory system in which intervals of a linear address space are allocated and de-allocated dynamically. A situation may arise where there exists a substantial amount of free space, but its utility is low because it consists of a large number of small intervals. This condition is generally referred to as storage fragmentation, a problem which has received extensive investigation. References [1-4] provide an entry into the literature. Approaches to the problem generally fall into two categories: storage allocation methods and techniques for reorganization. Work on reorganization techniques for fragmentation reduction appears to have been motivated primarily by questions related to the use of main or random access memory. Relatively little attention appears to have been devoted to reorganization in the context of secondary storage management.

This problem, although conceptually similar to that for main storage, is actually substantially different because of the way disk space is used. Occupancy rates tend to be high, residence times for data sets are typically measured in days rather than seconds, and storage management is done on a daily or weekly basis. Individual records generally occupy a small fraction of the space, so that a free interval of one or two percent of the overall volume, which may be an unusable fragment in the main memory context, is here actually rather large. Moreover, system integrity requirements impose restrictions on how data items are moved.

This paper treats the problem of disk reorganization in the context of storage management for the IBM MVS operating system. Disk space is provided by devices (disk drives) such as IBM 3330-11's. Each contains several *volumes* or *packs*, which for 3330-11's consist of what we regard as a linear space of tracks T_i, $i = 1, 2, \cdots, 15\,352$, with approximately 13 000 bytes/track.

Entities stored in this space are termed *data sets*, each of which is generally stored on a single pack. The minimum unit of space allocation is a single track. Space allocation is subject to the MVS rules, which are approximately as follows: The first space request associated with a given data set must be satisfied with no more than five *extents*, where an extent is a contiguous set of tracks T_j, $T_{j+1}, \cdots, T_{j+g}$. Subsequent requests may be satisfied by allocating additional extents, but the maximum number of extents per data set is sixteen. If a request cannot be granted, the associated task is abnormally terminated (ABENDed).

An extent may be reclaimed (de-allocated), as the result of user action or by the system space management. In the latter case, unused allocated extents may be reclaimed, or entire data sets may be migrated (shifted to a slower medium, such as tapes).

Space management is generally invoked on a daily or weekly basis, with the objective of providing sufficient

space until the next scheduled invocation [5]. Specific criteria for migrating data sets may vary by installation, but the following is not untypical. Data sets are migrated until a certain threshold of occupation is reached, for example, no more than 85% of the volume occupied. In addition, data sets with an excessive number of extents (more than eight, for example) are migrated and scheduled for automatic restoration. This process reduces the number of extents, leaving more potential for expanding data sets without exceeding the extent limit of 16. At this point, a determination is made of the usability of the resulting free space. If the available space is overly fragmented, the alternatives today are to either migrate additional data sets or to copy the contents of the volume. The former approach is generally not desirable because the migrated data sets may soon be referenced. The latter alternative, copying the pack in order to coalesce the free space, is what is often done in practice. A large MVS installation may have a schedule for copying the contents of several disks nightly. This is an expensive operation, requiring a substantial amount of machine time, system re-IPL, as well as manual handling of packs.

A possible alternative to either additional migration or volume copying is a partial reorganization in place. Such an operation might be advantageous if a substantial improvement in the free space distribution is obtained at the expense of moving a small fraction of the disk contents. Since copying is an operation which due to its expense can be done only infrequently for each disk pack, partial reorganization might yield an improved average space distribution, with fewer extents per data set. This could result in decreased arm travel for batch operations, as well as fewer job ABENDs due to exceeding the allowable number of extents. An additional advantage is the possibility of reorganizing disk packs in parallel with normal operation.

The objective of a reorganization is to improve the usability of the available free space. How then can one tell whether one space distribution is superior to another? This is a question which has not been fully answered, although measures of fragmentation have been proposed [6]. However, decisions on whether to copy a volume are often made on the basis of histograms of free interval sizes. The approach taken below is to attempt to move data so as to increase the amount of available space in large free intervals.

The following is a synopsis of the paper. The approach to reorganization is described in Section 2. This consists of a procedure for choosing a set of extents, then moving this chosen set to new locations in the volume (the terms pack and volume will be used interchangeably below). Section 3 considers the construction of selection functions, whose objective is to choose extents whose current locations are judged to be least advantageous. Sections 4 and 5, respectively, describe the use of selection functions to pick relocatable extents and the movement of these extents to new locations. Section 6 considers the possibility of performing iterative reorganization, that is, carrying out the above procedures more than once. The results of experiments on a number of volumes from the IBM Research Center, Yorktown, are given in Section 7. Substantial improvements were obtained in the free space distribution at the expense of moving only a fraction of the volume contents. Analysis of a simple model, described in the appendix, suggests that the favorable results should not be surprising.

2. Outline of the method

The set of extents and free spaces occupying a volume is regarded as a sequence of spaces S_i, $i = 1, 2 \cdots, M$, where S_i is contiguous to S_{i-1} and S_{i+1}. Each S_i may be either a free space or an extent. An *interval* $I_{j,k}$ is the smallest contiguous set of tracks containing S_j and S_k, $j \leq k$. If both S_j and S_k are free spaces, then $I_{j,k}$ is termed a *proper* interval. The number of occupied tracks in an interval and the length of the interval are denoted by $L_{j,k}$ and $M_{j,k}$, respectively.

The objective of reorganizing the pack is to improve the usability of the available free space, decreasing the number of small free intervals. Clearly there are many possible ways to do so. For example, a list could be made of all free spaces, followed by an attempt to find extents to fill the smallest ones. Most such approaches run into the problem of having to consider large combinations of extents and free intervals, resulting in substantial computational complexity.

The method adopted here simplifies the computation by separating the reorganization process into two essentially independent phases. The first results in the selection of a set of proper intervals (defined above). Extents residing in these intervals are termed *chosen* or *movable*.

In the second phase, an attempt is made to move the chosen extents to new locations. The intended effect is twofold: Chosen intervals are cleared and their original contents used to fill small fragments of available storage elsewhere in the pack.

The restriction to proper intervals was instituted to restrict the amount of computation. The overall number of intervals and the number of proper intervals are, respectively, approximately proportional to the square of the number of spaces (of which there might be several thou-

sand) and the square of the number of free spaces (of which there might be a few hundred).

Interval selection is carried out through the use of two measures G and C defined on the intervals. $G_{j,k}$ represents what might be considered an estimated gain or benefit from attempting to clear $I_{j,k}$, and $C_{j,k}$ is the cost of moving its contents.

3. Cost and gain functions

A reasonable measure of the cost of choosing $I_{j,k}$, conforming to common device characteristics, is $M_{j,k}$, the number of occupied tracks in the interval.

$$C_{j,k} = M_{j,k}. \tag{1}$$

This is the cost function which will be assumed throughout the paper.

Factors which affect the gain obtained from choosing an interval $I_{j,k}$ include:

1. The size (number of tracks) of the interval, $L_{j,k}$.
2. The sizes of the individual extents in the interval. It is preferable to move small extents, which might be used to fill fragments of available storage.
3. The distribution of extents within the interval, i.e., a measure of how these extents fragment the interval.
4. Statistics describing the free space and extent size distributions.
5. The algorithm for relocating chosen extents.

Before discussing specific gain functions, it is convenient to develop some additional notation and definitions.

Let $V_{j,k} = (q_1, \underline{m}_1, \underline{m}_2, q_2, \cdots)$ be a tuple associated with the interval $I_{j,k}$. The quantity q_i denotes the number of tracks in the ith free space in the interval counting from the leftmost or lowest numbered track. Similarly, $\underline{m}_i$ represents the number of tracks occupied by the ith extent.

As an example, consider an interval $I_{3,5}$, whose spaces S_3, S_4, and S_5 are, respectively, a free space of five tracks, an extent occupying eight tracks, and a free space of three tracks. Then $V_{3,5} = (5, \underline{8}, 3)$. Underlined numbers represent extents.

Definition
An *allowable* insertion of an extent is one which places an extent into either a left or right justified position in a free space.

Definition
An operation $O(I_{i,k})$ on an interval $I_{j,k}$ is a sequence of removals and allowable insertions of extents into the interval.

The result of an operation is a new numbering for the spaces $\{S_i\}$ in the pack and a new set of intervals. Note that if the interval $\{I_{j,k}\}$ is proper, the result of the operation on $I_{j,k}$ is an interval in the new ordering, $I'_{e,m}$. That is, the boundaries of $I_{j,k}$ are boundaries for spaces in the new ordering. An operation on a proper interval $I_{j,k}$ with its associated $V_{j,k}$ thus yields an $I'_{e,m}$ with an associated $V'_{e,m}$.

Definition
A clearing operation changes $V_{j,k}$ to $V'_{c,e} = (q_1)$. That is, all extents are removed.

An operation $O(I_{j,k})$ may be performed as a sequence of $\{O_i(I_{m,n})\}$ of operations on proper subintervals $\{I_{m,n}\}$ of $I_{j,k}$.

Definition
Associated with an operation $O(I_{j,k})$ on proper interval $I_{j,k}$ is a gain $G(V_{j,k}, V'_{e,m})$, where $V'_{e,m}$ is the resulting interval. If O is the clearing operation, this is denoted simply by $G(V_{j,k})$ or $G_{j,k}$.

A concept one might associate with a gain function is that equivalent changes should provide equivalent gains. One way of formalizing this notion is the following:

Definition
A gain function for an operation $O(I_{j,k})$ is termed *regular* iff $G(V_{j,k}, V'_{j',k'}) = \Sigma G_i(V_{m,n}, V_{m',n'})$ for any sequence $\{O_i(V_{m,n}, V_{m',n'})\}$ of proper subintervals $I_{m,n}$ of $I_{j,k}$.

As an illustration of the notion of regularity, consider the intervals shown in Fig. 1. Suppose $I_{1,3}$ is chosen, i.e., a decision is made to clear this interval. How then should one regard $I_{1,5}$? Given the decision, the difference between clearing $I_{1,3}$ and $I_{1,5}$ might be considered equivalent to clearing an interval such as $I_{30,32}$ which is identical to that resulting from the removal of m_1 from $I_{1,5}$. Clearing $I_{1,5}$ would then be equivalent to clearing $I_{20,22}$ and $I_{30,32}$. Note that the resulting change in the distribution and the extents to be moved are the same.

Regularity may, however, not always be appropriate. This is partly because the selection and relocation phases of the reorganization are independent, so that the operation which will be carried out on a particular interval is not precisely determined in advance. Suppose for example that the intervals discussed above are not cleared, but rather have their contents moved to a left justified position. Here the results of choosing $I_{1,5}$ would not be identical to those obtained from $I_{20,22}$ and $I_{30,32}$.

It can be shown that a sufficient condition for a function G to be regular for the clearing operation is that it be of the form

$$G_{j,k} = H(L_{j,k}) - \Sigma H(q_i) - \Sigma P(m_i) \qquad q_i, m_i \in V_{j,k}. \quad (2)$$

The above provides a valuation of free intervals which is a function $H(\cdot)$ of their size. $P(m_i)$ may be regarded as the expected reduction in the value of the overall free space resulting from the reinsertion of an extent of size m_i.

Two regular gain functions were used in the experiments. Both have $H = P$, with $H(x) = x^2$ and $x \log x$, respectively. These tend to favor large intervals with low occupancy and a few large free spaces. For example, an interval with $V_{i,j} = (50, 2, 50)$ would be favored over ones with V's given by (1, 2, 99) and (25, 1, 25).

A number of irregular gain functions were also considered with the one chosen for the experiments given by

$$G_{j,k} = \frac{L_{j,k}^2 - \Sigma q_i^2 - \Sigma m_i^2}{L_{j,k}}. \quad (3)$$

This produces a ratio $R = G/C$ with the following properties:

1. It is dimensionless. This may be desirable because volumes in practice exhibit different usage patterns. Some have relatively large and others relatively small extents and free spaces.
2. R is upper bounded by the inverse of the space utilization within the interval.
3. R tends to increase with the level of fragmentation within the interval. For example, location of a single extent in the center produces a larger value than one near the edge.

The above G thus tends to favor badly fragmented intervals with low average utilization and with small extents. Experimental results suggest that it leads to the choice of those intervals which might be picked from a visual inspection of the space allocation map.

4. Interval selection

Once C and G functions have been defined, an attempt can be made to maximize the overall gain, possibly subject to some limit on the expected cost. This section discusses some simple procedure for obtaining sub-optimal sets of intervals, based on choosing those with sufficiently large benefit/cost ratios.

Algorithm 1

Let ψ be a threshold, C_σ a cost limit, and Q a set of chosen intervals. Let $I_{j,k} \cap Q$ denote the intersection of $I_{j,k}$ with the union of intervals in Q. Let $I_{j,k} \cap^* Q$ be the set of intervals in Q which intersect $I_{j,k}$ and $I_{j,k} \cup [I_{j,k} \cap^* Q]$ be the interval obtained by the union of $I_{j,k}$ and those members of Q which intersect $I_{j,k}$ ($I_{j,k}$ is assumed to be proper). Initially, Q is empty.

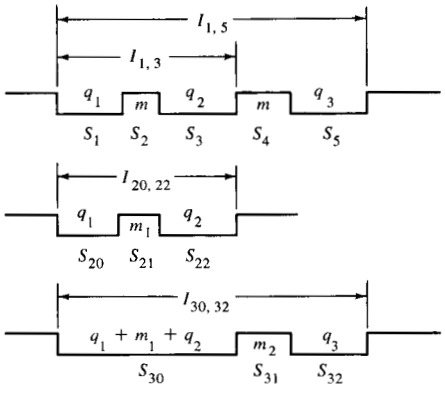

Figure 1 Intervals comprised of extents and free spaces.

For $j = 1, 2, \cdots$

$i = j - 1, j - 2, \cdots, 1.$

For proper $I_{i,j}$, such that $M_{i,j} \neq 0$,

If $I_{i,j} \cap Q = \varnothing$, then

If $R_{i,j} \geq \psi$, and

$C_{i,j} < C_\sigma$,

add $I_{i,j}$ to Q.

If $I_{i,j} \cap Q \neq \varnothing$, then

If $I_{i,j} \cup [I_{i,j} \cap^* Q]$ has a $G/C \geq \psi$

and $C < C_\sigma$,

remove $I_{i,j} \cap^* Q$ from Q and add

$I_{i,j} \cup [I_{i,j} \cap^* Q]$. □

Algorithm 1 produces a set of disjoint proper intervals each of which has an $R \geq \psi$, where ψ may be chosen to restrict the overall cost. It should be noted that this set is not necessarily unique; two proper intervals $I_{j,k}$ and $I_{e,m}$ may each have an associated $R \geq \psi$, but their union may not. The choice of one may thus preclude the choice of the other.

In the case of G regular [with C given by (1)], the problem of interval selection is simplified by the fact that many suitable G's (i.e., gain functions which are easy to compute and which yield desirable interval choices) tend to produce ratios R which grow with the interval size. To see why this is so, note that if the pattern of free spaces and extents is roughly uniform, the quantities $\Sigma H(q_i)$, $\Sigma P(m_i)$, and C will tend to be proportional to the interval size. Thus, if $H(L)$ grows faster than L, so does R. On the

other hand, $H(L) = KL$ is not suitable, since

$$G_{j,k} = K[L_{j,k} - \Sigma q_i] - \Sigma P(m_i) \qquad m_i, q_i \in V_{j,k} \tag{4}$$

is independent of the location of extents within the interval. This would result, for example, in intervals with associated V's of (1, 1, 100) and (50, 1, 51) being judged equally desirable for selection.

The result of the growth of R with the interval size is that Algorithm 1 tends to choose a single interval. This was confirmed by experiments using the regular gain functions of the previous section, resulting in the adoption, for these functions, of a procedure which simply chooses that proper interval with the largest value of G subject to $C \leq C_\sigma$. This will be referred to as Algorithm 2.

5. Relocation of chosen extents

Phase 2 of the procedure assigns locations to the chosen extents and constructs a schedule for moving them. Two essentially different approaches are: a) migration of the extents to another medium, followed by reinsertion, and b) reorganization in place. The latter alternative was chosen so as to avoid any requirements for manual intervention and also to permit movement of the extents during normal system operation.

Movement of an extent is constrained by the data integrity requirements of the system. Specifically, this means that an extent must be fully replicated before the original is erased, so that a valid copy is always available in case of system failure.

Let the set of chosen extents be described by $\{E_j\}$, $j = 1, 2, \cdots, n_c$. Relocating these extents involves a number of operations. Space must be allocated according to some schedule $T_{j(i)}$, $i = 1, 2, \cdots, n_c$, from the free space list which is obtained from Phase 1. A schedule must then be constructed which describes the order in which the chosen extents are moved.

More precisely, a description of Phase 2 requires:

1. Specification of the schedule $T_{j(i)}$. This might for example be largest or smallest first.
2. The assignment function which determines which free space is allocated to a given interval (e.g., best fit or first fit [1, 4]).
3. The effect on the free space list of assigning a location to an extent. For example, the resulting fragment could be added to the list.
4. The schedule for moving extents to new locations.

The following algorithm provides an assignment of extents to free spaces which are not contained in intervals chosen in Phase 1. The goal is to eliminate small free intervals and thus to effect a further shift of available space to large intervals. The algorithm is based on a best fit criterion, with space assigned to extents in order of decreasing size. Thus fragments created on insertion can be filled by small extents, many of which typically occupy one or two tracks.

Algorithm 3

Chosen extents are sorted by size and denoted by $\{E_j\}$, $i_j = 1, 2, \cdots, n_c$, where $i < j$ implies $L(E_i) \geq L(E_j)$. Let S denote a set of free spaces not included in the set of chosen intervals. Initially, S includes all such spaces.

1. Set $j \leftarrow 0$.
2. $j \leftarrow j + 1$.
   If $j > n_c$, the procedure terminates.
3. Find a free interval in S which provides a best fit for E_j, and remove it from S. If no such interval exists, go to 2.
4. Move the extent into the lowest numbered tracks in the space obtained by 3.
5. Add the fragment resulting from insertion of E_j to the free space list S.
6. Go to 2. □

The overall effect of the reorganization can be improved by attempting to move to a left justified position within their chosen intervals those chosen extents not successfully relocated by Algorithm 3.

Algorithm 4

1. Perform the operations indicated in Algorithm 3.
2. Let $E_i(j, k)$ be the set of extents remaining in $I_{j,k}$ after 1; $i < n$ implies $E_i(j, k)$ is to the left of $E_n(j, k)$.
3. For $i = 1, 2, \cdots$, move $E_i(j, k)$ to the leftmost feasible position in $I_{j,k}$. This is a position which may have been cleared by moving a lower-numbered extent, but which does not overlap the current location of $E_i(j, k)$.

□

6. Iterated reorganization

Application of the above techniques results in a new organization of the volume contents. Let A_0 and A_1 denote respectively the original and new organizations, and ζ the reorganization process:

$$\zeta(A_0) = A_1. \tag{5}$$

Definition

A reorganization process ζ is termed *final* if

$$\zeta^2(A_0) = \zeta(\zeta(A_0)) = \zeta(A_0) \tag{6}$$

for any initial organization A_0.

Consider the insertion of an extent at the left edge of a free space S_i. Let S_i' denote the remainder of this space after insertion (if the remainder is empty S_i' will denote the right boundary of S_i). The ratio $R = G/C$ and the gain function G are termed *nonincreasing on insertion* (NONI) if the ratio R associated with each proper interval containing S_i is no smaller than that for the corresponding proper interval containing S_i'.

Intuitively, the NONI property may be expected to hold for suitable gain functions, since insertion of an extent decreases the number of free tracks without increasing the number of free spaces. Examples of gain functions which yield NONI ratios [with C given by (1)] are (3) and the regular C's resulting from $H = P$ with $H(x) = x^2$ or $x \log x$.

Proposition 1
Suppose ζ is the application of Algorithms 1 and 4 with $C_\sigma = \infty$, G is regular, and R NONI. Suppose further that all chosen extents are relocated. Then ζ is final.

Proof
Reorganization may be viewed as the result of two operations: a) the removal of all chosen extents and b) their reinsertion.

It is first shown that no nonempty proper intervals with $R \geq \psi$ remain after a). Suppose there is such an interval $I'_{j,k}$, with $R'_{j,k} \geq \psi$. Since all chosen extents are removed, $I'_{j,k}$ must include some chosen subset $\{I_{e,m}\}^c$ of proper intervals in the original organization. Let C_0 and C_1 denote, respectively, the number of occupied tracks in $\{I_{e,m}\}^c$ and $I'_{j,k}$. Let $I_{q,r}$ denote the interval $I'_{j,k}$ before the removal of the extents contained in $\{I_{e,m}\}^c$.

The regularity of G implies that

$$G_{q,r} = G'_{j,k} + \Sigma G_{e,m} \qquad I_{e,m} \in \{I_{e,m}\}^c, \tag{7}$$

but

$$\frac{G'_{j,k}}{C_1} \geq \psi, \tag{8}$$

and

$$\frac{\Sigma G_{e,m}}{C_0} \geq \psi. \tag{9}$$

Thus

$$\frac{G_{q,r}}{C_{q,r}} = \frac{G'_{j,k} + \Sigma G_{l,m}}{C_0 + C_1} \geq \psi, \tag{10}$$

so that $G_{q,r}$ would have been chosen; $R_{q,r} \geq \psi$. No proper intervals remain with cost benefit ratios $\geq \psi$ after (a).

The NONI property insures that no such intervals remain after (b). □

If $C_\sigma < \infty$, the reorganization will in general not be final, even for regular G and R NONI. It is not difficult to show, however, that intervals in A_1 (the resulting organization) for which $R \geq \psi$ and $C \leq C_\sigma$ intersect those chosen in the selection phase of ζ.

Suppose reorganization is to be performed with a limit of $2x$ tracks to be moved. Is it better to reorganize in two passes, each moving x tracks, or in a single pass with a limit of $2x$? If G is regular, the gain (as measured by G) from the choice of the two passes is the same as that obtained by a single pass using the same chosen intervals. Moreover, the above argument suggests that the choice for two passes will tend to be more restrictive. Thus, if G is regular and a good measure of the expected gain, a single pass may be superior.

Suppose G is irregular. Then the initial reorganization may create some new intervals with high values of R. That is, the first pass may create additional opportunities. This was found to be the case in the experiments.

7. Experimental results

This section describes the outcome of experiments on the contents of a number of disk packs at the IBM Thomas J. Watson Research Center, Yorktown Heights, NY. Results are given for four representative volumes.

Let $f(x)$ denote the fraction of free space comprised of intervals no larger than x. Similarly, let $u(x)$ be the fraction of occupied space (excluding one-time allocation for permanent data sets) contained in extents no larger than x. The function $u(x)$ may be regarded as predictor of storage requests for the pack.

The initial space and utilization distribution and the results of reorganization for each of the packs are illustrated in a set of figures.

Figures 2–8 show for volumes 1 to 4, respectively, the following curves:

1. $f_0(x)$, the distribution of free space before reorganization. Numbers at the right-hand side of the figures indicate the size of the largest free space.
2. $f_1(x)$, the result of a reorganization ζ_1, which consists of the application of Algorithms 1 and 4 with a threshold $\psi = 2$ and C_σ set to Q, the total number of free tracks on the volume. G is the irregular function (3) described in Section 3.

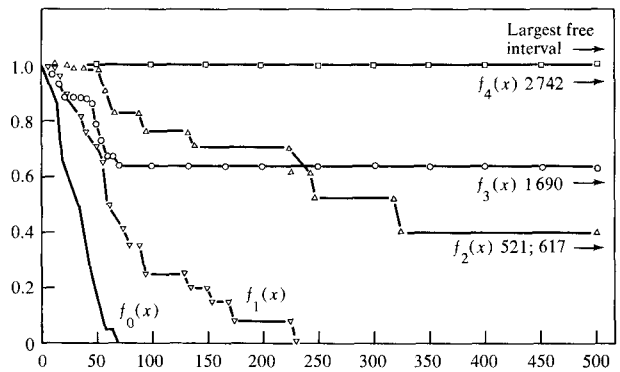

Figure 2 Distribution of free space for volume 1 (12 610 occupied, 2742 free tracks). Number of tracks moved—$f_1(x)$: 478; $f_1(x)$ and $f_2(x)$: 1250; $f_3(x)$: 1228; $f_4(x)$: 2722.

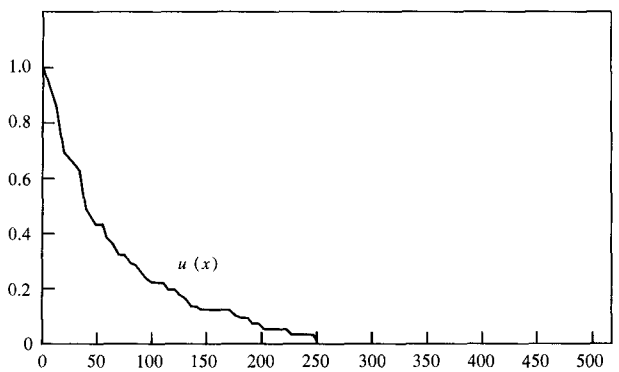

Figure 3 Fraction of occupied space $u(x)$ contained in extents no larger than x for volume 1. (Tracks occupied, 12 610, 11 145 for user data.)

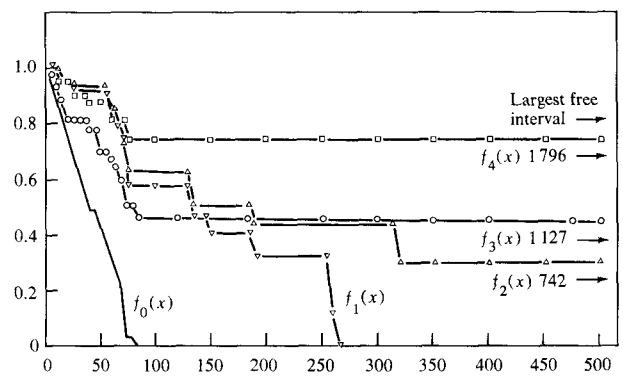

Figure 4 Distribution of free space for volume 2 (12 922 occupied, 2430 free tracks). Number of tracks moved—$f_1(x)$: 576; $f_1(x)$ and $f_2(x)$: 901; $f_3(x)$: 851; $f_4(x)$: 1990.

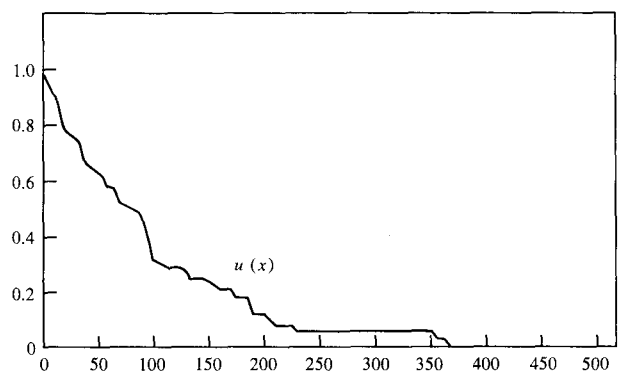

Figure 5 Fraction of occupied space $u(x)$ contained in extents no larger than x for volume 2. (Tracks occupied, 12 922, 11 457 for user data.)

3. $f_2(x)$, the result of applying a second iteration of ζ_1 to the result obtained in (2). If A_0 represents the original organization, $f_2(x)$ is the free space distribution for $\zeta_1(\zeta_1(A_0))$.
4. $f_3(x)$, the result of applying a transformation ζ_3 consisting of Algorithms 2 and 4 with C_σ equal to the number of tracks moved in the two iterations of ζ. G is given by

$$G = L_{j,k} \log L_{j,k} - \Sigma q_i - \Sigma m_i \log m_i$$

$$q_i, m_i \in V_{j,k} \qquad (11)$$

This yields a comparison of the two gain functions for similar numbers of tracks moved.

5. $f_4(x)$, the result of using ζ_3 with $C_\sigma = Q$ and G as in 4.

Figure 3 shows $u(x)$ for volume 1. Results for volumes 2, 3, and 4 are given in Figs. 5, 7, and 9, respectively.

Let

$$Y_i = \max [u(x) - f_i(x)]. \qquad (12)$$

Suppose space is to be allocated for a set of extents whose size is distributed according to $u(x)$ with a total requirement of Q tracks. Then Y_i gives a lower bound for the fraction of space that cannot be allocated if the free space distribution is f_i.

The results indicate that ζ_1 generally yields a substantial improvement in the free space distribution at the cost of moving only a few percent of the volume contents. A second iteration of ζ_1 yields a free space distribution which results in $Y_2 = 0$ for all packs. Application of ζ_3 with a cost limit resulting in the relocation of a number of tracks similar to that resulting in f_2 generally yields less reliable results: $Y_3 > 0$ for volumes 3 and 4. However, if

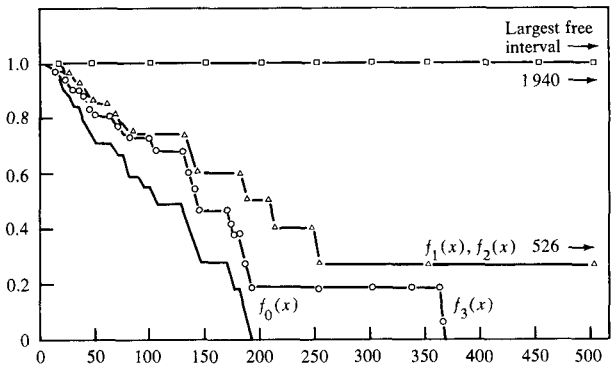

Figure 6 Distribution of free space for volume 3 (13 412 occupied, 1940 free tracks). Number of tracks moved—$f_1(x)$: 260; $f_1(x)$ and $f_2(x)$: 260; $f_3(x)$: 244; $f_4(x)$: 1843.

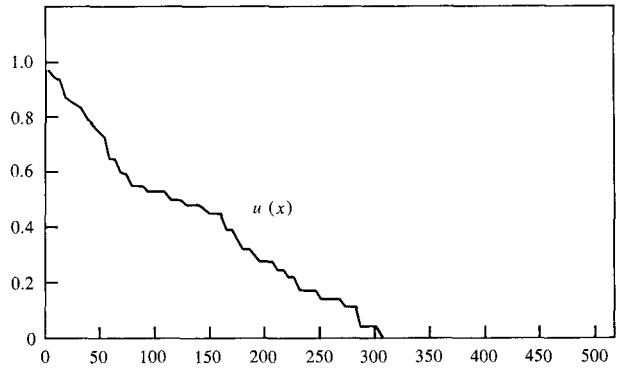

Figure 7 Fraction of occupied space $u(x)$ contained in extents no larger than x for volume 3. (Tracks occupied, 13 412, 8187 for user data.)

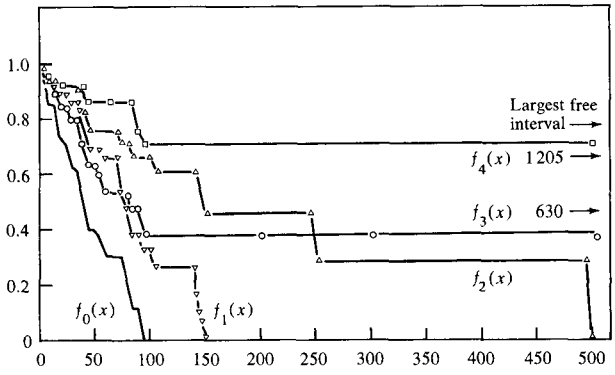

Figure 8 Distribution of free space for volume 4 (13 629 occupied, 1723 free tracks). Number of tracks moved—$f_1(x)$: 219; $f_1(x)$ and $f_2(x)$: 497; $f_3(x)$: 488; $f_4(x)$: 1226.

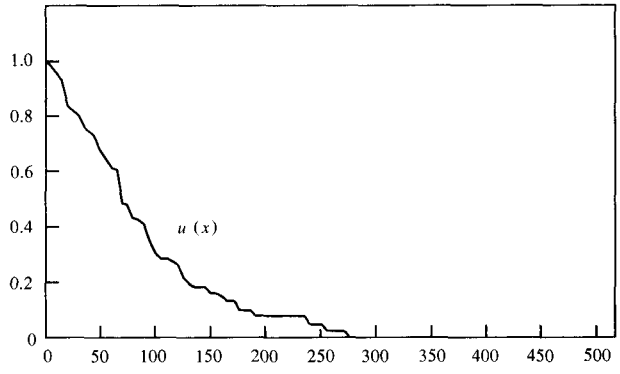

Figure 9 Fraction of occupied space $u(x)$ contained in extents no larger than x for volume 4. (Tracks occupied, 13 629, 9829 for user data.)

the cost limit is relaxed, the resulting organization contains in all cases a free interval of size comparable to the overall free space.

8. Discussion and conclusion

A technique was described for partially reorganizing the contents of disk storage so as to reduce the level of fragmentation. Experiments suggest that the method may be used to substantially improve the utility of the available free space at the expense of moving only a fraction of the disk contents.

Two types of functions were considered for use in the process of selecting intervals to clear. The first has the tendency to choose a single interval, while the second chooses a multiplicity. For a limited number of tracks to be moved, the second appears somewhat superior, since the choice of a single interval often appears to result in the increased fragmentation of the remaining space. A decision between one or the other, however, should probably be dictated by implementational considerations.

Given the technique for increasing the utility of the free space, a further possibility is the coalescing of data sets which initially occupy several extents. This could be done by using the free space obtained by the reorganization for coalescing data sets which occupy more than a given number of extents, followed by further reorganization. This process could be iterated if necessary. Note that since the reorganization may be done by moving on the order of ten percent of the volume contents, the result could be essentially equivalent to a complete copy of the volume at a fraction of the expense in terms of the number of tracks moved. An additional advantage is the avoidance of the requirement for manual handling of disk packs.

Appendix: A simple model

Experimental evidence indicates that reorganization generally leads to a substantial improvement in the free space distribution. An interesting question is whether this is what should be expected. In the following, the results of an analysis of a simple model suggest that positive experimental outcomes should not be surprising.

The model is as follows. Storage space is assumed to be an infinite sequence of slots each of which is occupied with probability p and free with probability $q = 1 - p$. The value of p was chosen as 5/6, yielding a utilization of 0.83, or roughly that encountered in practice.

Contents of an occupied slot are moved if and only if it is bordered on either side by a free space. Note that this tends to result in choosing fewer candidates for movement than a more realistic criterion, which would not arbitrarily exclude extents occupying more than a single slot.

The above choice of intervals corresponds to a regular gain function whose value is the number of occupied slots bordered on each side by free intervals, a cost given by the number of occupied slots, and a choice threshold of one.

The probability that a free interval is of size n slots is given by

$$P_f(n) = pq^{n-1} = (5/6)\ (1/6)^{n-1}. \tag{A1}$$

A randomly chosen free slot is bordered on each side by an occupied slot with probability $(5/6)^2$. Thus $Q = (5/6)^2$ of the free space consists of intervals of length one. It can be shown that the percentage of free space occupied by free intervals of length 2 is $Q_2 = (1/3)\ (5/6)^2$. In other words $25/27 \approx 0.93$ of the free space before reorganization consists of intervals of size two or smaller.

The probability that a free interval is affected by the reorganization (i.e., borders a slot whose contents will be moved) is

$$V = 2pq + q^2 = (11/36). \tag{A2}$$

Moreover, this is independent of the interval size.

The number of slots whose contents will be moved is lower bounded by one half the number of free intervals affected. In other words, at least 0.153 times as many extents of size one will be moved as there are free intervals.

After reorganization (using a best fit criterion), the proportion of free space in intervals of length 1 is

$$Q' \geq \left[\frac{P_f(1)[1 - V] - V/2}{P_f(1)}\right] Q_1 \approx 0.34. \tag{A3}$$

The proportion of free space in intervals of length 2 is

$$Q_2' = (1 - V)Q_1 \approx 0.16. \tag{A4}$$

In other words, the proportion of free space in intervals of length 3 or greater is changed from about 0.07 to more than 0.50. This is done by moving about 0.025 of the overall contents.

Acknowledgment

The authors acknowledge valuable discussions with N. Pass, who suggested this area of investigation, as well as the assistance of D. Brinkley in the preparation of the manuscript.

References

1. D. E. Knuth, *The Art of Computer Programming*, Vol. 1, Addison-Wesley Publishing Company, Reading, MA, 1968.
2. J. M. Robson, "Bounds for Some Functions Concerning Storage Allocation," *J. ACM* **21,** No. 3, 491–499 (1974).
3. L. P. Deutch and D. G. Bobrow, "An Efficient, Incremental, Automatic Garbage Collector," *Commun. ACM* **19,** No. 9, 522–526 (1976).
4. J. E. Shore, "On the External Storage Fragmentation Produced by First-Fit and Best-Fit Allocation Strategies," *Commun. ACM* **18,** No. 8, 433–440 (1975).
5. J. P. Considine and J. J. Myers, "MARC: MVS Archival Storage and Recovery Program," *IBM Syst. J.* **16,** No. 4, 378–397 (1977).
6. J. P. Considine, "A Computable Measure of Fragmentation for Direct-Access Volumes," *Research Report RC6241 (#26807)*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1976.

Received August 11, 1978; revised November 6, 1978

The authors are located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.