Worst case fragmentation of first fit and best fit storage allocation strategies

J. M. Robson

Department of Computer Studies, University of Lancaster, Bailrigg, Lancaster LA1 4YN

The worst possible storage fragmentation is analysed for two commonly used allocation strategies. In the case of the first fit system, fragmentation is not much worse than is inevitable but for the best fit system, it is almost as bad as it could be for any system.

(Received January 1975)

1. Introduction

When a dynamic storage allocation system is used with limits on the size of blocks allocated (n) and the total amount of store busy at any time (M), the store size necessary to guarantee against breakdown due to fragmentation is a function of M and n.

It has been shown (Robson, 1974) that for an optimal strategy, this function lies between $\frac{1}{2}M \log_2 n$ and about $0{\cdot}84M \log_2 n$ asymptotically but practical allocation strategies chosen for their ease of implementation may require more space than this.

For the buddy system (Knowlton, 1965) it can be shown that about $2M \log_2 n$ words are sufficient (Knuth, 1973). This paper gives results for the first fit and best fit systems.

The first fit system turns out to be not far from the optimum. About $M \log_2 n$ words of store are sufficient. On the other hand the best fit system needs about Mn words.

These results should be considered in conjunction with the simulations of Shore (1975) which suggest that even when an allocation system is run continuously on the brink of breakdown due to fragmentation, memory utilisation averages about 70% to 95% for a fairly wide range of distributions of allocated block size. Clearly the sort of catastrophic fragmentation here shown to be possible occurs only very rarely.

2. The worst worst case

The worst fragmentation that any system can encounter (unless like the buddy system it can break down although a gap large enough to meet the request exists) is, provided $M \geqslant 2n$, a sequence of $(M - n)$ single word blocks each preceded by a gap of $n - 1$ words. This is illustrated in **Fig. 1.**

If this pattern is allowed to occur, then $n(M - n) + n - 1$ words will not be enough to prevent breakdown if a block of n words is required. It may be noted that one recommended strategy, the 'modified first fit' of Knuth (1973), does allow exactly this pattern to be produced (by the allocation of $n(M - n)$ single word blocks of which those lying in a gap of Fig. 1 are freed as soon as one more block has been allocated).

The analysis below of the first fit and best fit systems is a study of how the former avoids any pattern at all like that of Fig. 1 whereas the latter can be forced into one very similar to it.

3. The first fit system

The system analysed here is the one in which any request for a block of x words is met by using the first x words of the first gap of x or more words.

The reason why this system avoids the catastrophic fragmentation of Fig. 1 is that small blocks are always allocated near the beginning of store. It will in fact be shown inductively that all blocks of size up to j inclusive will be allocated in the first $\sum_{i=1}^{j} Z(i)$ words of store where $Z(i)$ is defined as $M/(i \ln 2)$.

Thus the worst possible fragmentation when a $j + 1$ word block is to be allocated consists of a sequence of areas of size $Z(i)$ each covered with a pattern of alternating i word blocks and j word gaps.

For $j = 1$ the assertion to be proved is trivial since $Z(1) > M$. Therefore suppose it proved for all values of j up to some size s. The inductive step proves it true also for $j = s + 1$.

The proof considers the situation where a block of $s + 1$ words cannot be allocated in the stated area and shows that if this is so, $M - s$ words must already be in use.

Define $D(j)$ as the distance from the start of the store to the end of the last block of j words or less.

Then $D(j) \leqslant \sum_{i=1}^{j} Z(i)$ by the inductive hypothesis $(1 \leqslant j \leqslant s)$.

In other words, if $\delta(j)$ is defined as $\sum_{i=1}^{j} Z(i) - D(j)$, then $\delta(j) \geqslant 0$ $(1 \leqslant j \leqslant s)$.

Since no block of $s + 1$ words can be allocated, the number of words already in use is more than

$$\frac{D(1)}{s+1} + \sum_{j=1}^{s-1} \frac{(D(j+1) - D(j))(j+1)}{j+s+1} + \frac{\left(\sum_{j=1}^{s+1} Z(j) - D(s) - (s+1)\right)(s+1)}{2s+1}$$

because the store contains:

1. An area of size $D(1)$ covered with blocks each preceded by a gap of s words or less.
2. A sequence of areas of size $D(j + 1) - D(j)$ covered with blocks of at least $j + 1$ words each preceded by a gap of s words or less.
3. An area of size $\left(\sum_{j=1}^{s+1} Z(j) - D(s)\right)$ covered with blocks of at least $s+1$ words, each preceded by a gap of s words or less of which the last may also be followed by a gap of less than $s + 1$ words.

But

$$\frac{D(1)}{s+1} + \sum_{j=1}^{s-1} \frac{(D(j+1) - D(j))(j+1)}{j+s+1} + \frac{\left(\sum_{j=1}^{s+1} Z(j) - D(s) - (s+1)\right)(s+1)}{2s+1}$$

Fig. 1

$$= \frac{Z(1) - \delta(1)}{s+1} + \sum_{j=1}^{s-1} \frac{(Z(j+1) + \delta(j) - \delta(j+1))(j+1)}{j+s+1} + \frac{(Z(s+1) + \delta(s) - (s+1))(s+1)}{2s+1}$$

$$\geqslant \frac{Z(1)}{s+1} + \sum_{j=1}^{s-1} \frac{Z(j+1)(j+1)}{j+s+1} + \frac{Z(s+1)(s+1)}{2s+1} - \frac{(s+1)^2}{2s+1} \text{ (since all } \delta(j) \geqslant 0)$$

$$> \sum_{j=0}^{s} \frac{Z(j+1)(j+1)}{j+s+1} - (s+1)$$

$$= \frac{M}{\ln 2} \sum_{j=0}^{s} \frac{1}{j+s+1} - (s+1)$$

$$> M - (s+1)$$

Thus any allocation which cannot be accommodated in the stated area is ruled out by the limit of M busy words. This completes the proof of the inductive step.

Putting j equal to n now gives the result that a store of $\frac{M}{\ln 2} \sum_{i=1}^{n} \frac{1}{i}$ words is sufficient for the first fit system.

Except for $n \leqslant 4$, this bound is less than the $M(1 + \log_2 n)$ established in (Knuth, 1973) for the buddy system with the assumption that all block sizes are powers of 2.

4. The best fit system

The best fit system always allocates any block at the start of the smallest gap large enough to accommodate it. This description is imprecise in that it does not specify which gap is used if several of the same size are available but no situations arise in this discussion where the ambiguity is important. It will be shown that this system needs about $M(n-2)$ words of store whenever $n > 5$. For $n \leqslant 4$ the same result follows from the lower bound given in (Robson, 1974) and for $n = 5$ it can be shown by an ad hoc argument which is not given here.

The proof shows that a pattern similar to the catastrophic fragmentation of Fig. 1 can be spread through almost $M(n-2)$ words of store. The pattern consists of one $n-2$ word block followed by any number of single word blocks each preceded by a gap of $n-3$ words. Once this pattern has been started it can be extended by the addition of one more single word block in the manner shown in **Fig. 2** for the addition of a fourth such block. First two $n-2$ word blocks are allocated (of which the second merely acts as a buffer) and then a sequence of freeings and allocations transforms the first of them into a $n-3$ word gap followed by a single word block. This distorts the pattern previously existing just to the left but this distortion then moves gradually leftward and finally disappears when it meets the initial $n-2$ word block. The details of this process are shown in Fig. 2.

Inspection of Fig. 2 will show that blocks will always be allocated in the position shown provided $2 < n - 3$ or in other words provided $n > 5$. Inspection of Fig. 2 also shows that the number of active words exceeds the initial value shown in the first line by at most $3n - 7$ so the process shown can be repeated until $M - (3n - 8)$ words have been used at which point the pattern covers $(M - (3n - 8) - (n - 3))(n - 2)$ words.

What has been demonstrated is that the best fit system will use at least $(M - 4n + 11)(n - 2)$ words if a sufficiently large store is available. The corollary that it will fail in a smaller store does not follow immediately because in such a store the process shown in Fig. 2 may be interrupted if a gap of $2n - 4$ words or less is left at the end of the store by the initial allocation of the two $n-2$ word blocks. To complete the proof it is also necessary to show that $3n - 7$ words may also cause the system to fail when the residual gap after the last 1 word block of the pattern is $4n - 8$ or less. This is established by the sequences of allocations shown in **Figs. 3** and **4** which each make impossible the allocation of a block of $n - 1$ words. The sequence shown in Fig. 3 will cause breakdown unless the residual gap was of $2n - 2$, $3n - 4$ or $3n - 5$ words in each of which cases the one in Fig. 4 will do so. Thus $2n - 1$ words can cause the system to fail when the residual gap is $4n - 8$ words or less and $2n - 1 \leqslant 3n - 7$ as required since $n > 5$.

This completes the proof that the best fit system needs a store of at least $(M - 4n + 11)(n - 2)$ words.

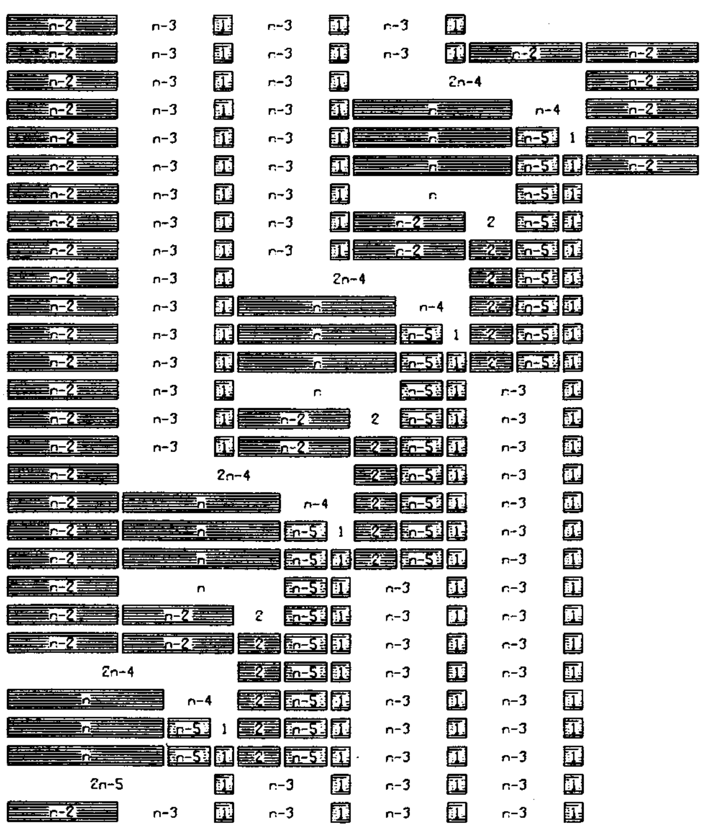

Fig. 2

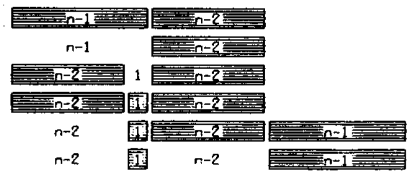

Fig. 3

Fig. 4

5. Summary

The worst possible fragmentation is considerable for all systems and has been shown to be much worse for the best fit than the first fit system. This contrasts with results derived from simulation which show that average fragmentation is small, roughly similar for first fit and best fit and sometimes better with best fit.

References

KNOWLTON, K. (1965). A Fast Storage Allocator, *CACM*, Vol. 8, pp. 623-625.

KNUTH, D. E. (1973). *Fundamental Algorithms*, Addison-Wesley.

ROBSON, J. M. (1974). Bounds for Some Functions Concerning Dynamic Storage Allocation, *JACM*, Vol. 21, pp. 491-499.

SHORE, J. E. (1975). On the External Storage Fragmentation Produced by First-Fit and Best-Fit Allocation Strategies, *CACM*, Vol. 18, pp. 433-440.

Foundations of Computer Science II, Parts 1 and 2, edited by K. R. Apt and J. W. de Bakker, 1976; 147 and 149 pages. (*Mathematical Tracts*, Dfl. 18 each)

This Tract contains the notes of five of the six series of lectures given at the Second Advanced Course on the Foundations of Computer Science organised by the Mathematical Centre in Amsterdam in June 1976.

The first (30 pages) is 'Graphical algorithms and their complexity' by E. L. Lawler. The author says that this is not intended as a survey but as a rather arbitrary and personal selection of problems intended to serve as an introduction to the methodology of the subject area. Problems considered are to find a 'topological ordering' of the nodes of an acyclic digraph, to recognise whether an acyclic digraph is 'series parallel', to find a polynomial time algorithm for the isomorphism problem restricted to series parallel digraphs, to find a minimum cost spanning tree in an undirected graph, to generate all maximal independent sets, to compute the chromatic number of a graph.

The second (106 pages, also in Part 1) is 'The complexity of data organisation' by J. Van Leeuwen. These concentrate on useful techniques in data organisation which can bring improvements in programs performing data manipulation. There are sections on 'Efficiency versus data representation', 'File-merging', 'Tables and balanced trees', 'Path compression', 'Associative search structures', 'Pattern matching'. Typical results are 'The two-tape merge procedure is stable and requires time linear in the length of the merged files', 'One can execute, find, insert, and delete instructions on an arbitrary HB-tree with N leaves in $\sim \log N$ steps per instruction' (Here an HB-tree is a binary search tree in which all leaves have equal depth and each node with only one son has a brother with two sons.)

The first two articles in Part 2 are 'Program semantics and mechanized proof' (47 pages) and 'Models of LCF' (19 pages) by R. Milner. The first of these takes a very simple language and studies its operational semantics (semantics by abstract machine, or by evaluation). Then using the work in the second paper it presents the denotational semantics in the style originated by Strachey and shows that the two semantic descriptions are equivalent in an appropriate sense. The second paper presents the model theory of a logic of computable functions, proposed by Dana Scott in 1969, in the form of a typed λ-calculus.

Next is 'L systems, a parallel way of looking at formal languages. New ideas and recent developments' (38 pages) by A. Salomaa. This reports on recent results in the rapidly growing subject of L systems. These were originally introduced by Lindemayer to provide mathematical models in biology and defined as linear arrays of finite automata. Later they were reformulated into the framework of grammar-like constructs and from then on their theory has developed essentially as a branch of formal language theory.

The last paper is 'Three hardest problems' by W. J. Savitch (38 pages). This is an introduction to complexity theory based on three illustrative examples: a hardest context free language, the set of codings of threadable mazes, the set of satisfiable Boolean expressions.

As is to be expected from the reputation of the authors, the articles are all excellent expositions and to be recommended as introductions to their subject matter.

J. C. SHEPHERDSON (Bristol)