A Model for an ISAM File with Multiple Overflow Chains

Y. MANOLOPOULOS*, D. KLEFTOURIS AND L. PETROU

Aristotelian University of Thessaloniki, School of Engineering, Faculty of Electrical Engineering, Department of Computer and Electronic Engineering, 54006 Thessaloniki, Greece

A model for an index sequential file employing multiple overflow chains per bucket is developed. This model is used to analyse the effects of insertions and deletions on the cost of successful and unsuccessful search in terms of block accesses. Numerical results are obtained illustrating the performance. The performance is also compared with that of an ISAM file using only one overflow chain per bucket.

Received July 1985, revised February 1987

1. INTRODUCTION

Index sequential files are widely used in various applications since they offer both direct and sequential processing. A problem inherent to their organisation is that after some time, due to insertions of new records, the main buckets of the file will overflow. This problem is handled by either using a splitting technique or chaining. Splitting is employed in VSAM files, while chaining is used in ISAM files.

Work on the performance evaluation of ISAM files is reported in Ref. 12–14. Until recently little has been published on dynamic files and even less on dynamic ISAM files. There are few attempts to evaluate the rate of increase of the search cost due to insertions and deletions. In Ref. 1 the performance of ISAM files is studied and evaluated as a function of time. However, this work is strictly deterministic since it assumes that the number of insertions and deletions per bucket is constant for all the buckets of the file. Larson[8] developed a stochastic model for the rate of insertions and deletions which is based on the 'birth-and-death' process. He also assumed that insertions obey a Poisson distribution, while the time interval in which a record is deleted obeys an exponential one. In a similar study Cooper[3] considered a continuous-parameter queuing model assuming that the records arrive in batches having a geometric distribution, while the time between two batch insertions obeys an Erlangian distribution. Heyman[5] employed a discrete-parameter random-walk model, assuming that the rate of deletions is fixed and independent of the number of records in the bucket. In all the above-mentioned studies it is assumed that the records have fixed length. However, in Ref. 2 various overflow handling techniques are studied assuming variable-length records.

In this paper we develop a model for ISAM files with multiple overflow chains for every main bucket. Using multiple overflow chains improves the performance of the ISAM file considerably. Consequently the expensive task of file reorganisation may be done less frequently. Reorganisation may also be postponed by batching request.[10] Note here that the idea of multiple overflow chains for every main bucket has been applied previously in an environment of hashed files.[9]

The paper is organised as follows. In Section 2 the particular chaining mechanism of the proposed model is described. In addition the probability distribution of the number of records per bucket is explained and the overflow searching cost is evaluated analytically for the cases of successful and unsuccessful search. Finally, in Section 3 some numerical results are obtained and the results are compared with the ones obtained from Larson's model.

2. DESCRIPTION OF THE MODEL

Assume that N records of fixed length are loaded into the file. The file contains buckets of fixed capacity b. The initial capacity of the main buckets is m records per bucket, where $m < b$, and thus free space is left in every main bucket. The keys of the records are modelled as mutually independent random variables having a distribution $F(K)$. The records are inserted into the buckets, where they are stored in an ascending order. The key interval assigned to the ith bucket is $(K_{(i-1)m}, K_{im}]$ and the probability that a record hits the ith bucket is $Q_i = F(K_{im}) - F(K_{(i-1)m})$. A static index is built up in the usual way. Since the index is not changed after loading, all the Q_is can be assumed fixed and equal to their mean value.[6] However, this assumption will be further discussed in Section 3.

Due to insertions and deletions of records, it is certain that after some time the free space will be exhausted in some of the main buckets. Consider the case of a full main bucket. Every new record arriving at this bucket is directed to the overflow area. In fact, if the key of the new record is lower than the key of the lth record and higher than the key of the $(l-1)$th record, then it initializes an overflow chain originated from the $(l-1)$th record of the bucket. In a similar manner every new overflow record either initializes a new overflow chain (up to b chains) or is placed in an already existing chain. The records in both the main bucket and the overflow chains are kept sorted according to their key values. Each record in the main bucket has a pointer indicating the first record in its associated overflow chain. The records in the overflow chains constitute linked lists. The NIL pointer of the last record of every overflow chain signifies return to the next record in the main bucket. Hence the proposed ISAM file model assumes one overflow chain for each record of the main bucket.

We assume that the overflow area is organised in the conventional fashion, i.e. some of the last tracks of every cylinder (the local overflow area), as well as some of the cylinders at the end of the file (the global overflow area) are left empty. It is assumed that all overflow records

* To whom correspondence should be addressed.

reside in the local overflow area and not in the global one.

For the sake of mathematical analysis we make the following assumptions. Deletions of records residing in the overflow chains are treated as simple deletions in linked lists. When a record of the main bucket is deleted and there is an overflow chain attached to it, the first record from the corresponding chain takes its place. If there is no overflow chain attached to it, the first record of any subsequent overflow chain enters the main bucket and all records between the deletion and insertion points are shifted. Thus the main bucket remains full and its records are kept sorted.

Solving the 'birth-and-death' stochastic process, an approximate probability distribution of the number of records per bucket at some point after loading is a steadily expanding modified Poisson distribution:[8]

$$P_m(x, \alpha) = 0 \qquad x = 0, 1, 2, \ldots, m-1$$
$$P_m(x, \alpha) = \frac{e^{-m\alpha}(m\alpha)^{x-m}}{(x-m)!} \quad x = m, m+1, \ldots \tag{1}$$

where m is the number of records per bucket at initial loading, x is the number of records per bucket at some point and α is the file expansion factor, which is defined as the ratio of the total number of records inserted in the file after the initial loading divided by the number of records that were initially loaded in the file.

Denoting by R_m, $m = 1, 2, \ldots, b$, the percentage of main buckets, which were loaded with m records initially, then the probability that a bucket contains x records at a file expansion factor equal to a is given by the formula:

$$P(x, \alpha) = \sum_{m=1}^{b} R_m P_m(x, \alpha) \quad x = 0, 1, 2, \ldots \tag{2}$$

In the following subsections we derive expressions for the expected number of accesses to the overflow blocks for the cases of successful and unsuccessful searches.

2.1 Successful search

The evaluation will take into account only the cost of search for overflow records. Consider the case when there are k overflow records attached to the ith bucket, $k > 0$. These k records are distributed over the b chains. For a sufficiently large b the appropriate probability distribution describing this process is the binomial one. According to this distribution the probability that a specific chain contains n records is given by the relation:

$$\binom{k}{n}\left(\frac{1}{b}\right)^n\left(\frac{b-1}{b}\right)^{k-n} \tag{3}$$

The probability that a successful search will hit a certain bucket depends on the number of records at some point after loading.[8] In other words it is more probable to hit buckets with many overflow records than ones with a few overflow records. This probability is equal to:

$$P(b+k, \alpha)(b+k)/X(\alpha) \tag{4}$$

where $X(\alpha)$ is the expected value of records associated with the corresponding main bucket at some point after loading, given by the following formula:

$$X(\alpha) = \sum_{x=1}^{\infty} xP(x, \alpha) \tag{5}$$

Given that a certain bucket is accessed, the probability of following a certain chain with n records depends on the length of the chain and is given by the relation (6):

$$\frac{n}{\dfrac{k}{b(1-(1-1/b)^k)}} \tag{6}$$

where $(1-(1-1/b)^k)$ is the probability that the length of an overflow chain is not zero and $k/b(1-(1-1/b)^k)$ is the expected value of the length of an overflow chain.

For the rest of the analysis we assume that either every overflow record of this chain resides on a different block or, equivalently, that the capacity of each overflow block is one record. The effect of these assumptions is discussed in Section 3. Consider now a chain of length n. Since any record of the file can be retrieved with equal probability, the first record is retrieved with one additional access, the second with two accesses, ..., the nth record with n accesses. Therefore the required additional accesses for the retrieval of all the overflow records of this chain are $1+2+\ldots+n = n(n+1)/2$. Thus the expected value of accesses for the retrieval of one overflow record is given by the following relation:

$$n(n+1)/2(b+k) \tag{7}$$

Taking into account that there are b overflow chains and summing relations (3–7) over x, k and n, the following equation is obtained:

$$S(\alpha) = \frac{b}{2X(t)} \sum_{k=1}^{\infty} P(b+k, \alpha) \sum_{n=1}^{k} \binom{k}{n}\left(\frac{1}{b}\right)^n \times \left(1-\frac{1}{b}\right)^{k-n} \frac{bn(1-(1-1/b)^k)}{k} n(n+1) \tag{8}$$

According to the binomial theorem[7] the second summation of (8) can be simplified as follows:

$$\sum_{n=0}^{k} \binom{k}{n}\left(\frac{1}{b}\right)^n\left(1-\frac{1}{b}\right)^{k-n} x^{n+1} = x\sum_{n=0}^{k} \binom{k}{n}\left(\frac{1}{b}\right)^n\left(1-\frac{1}{b}\right)^{k-n} x^n = x(1-p+px)^k \tag{9}$$

where $p = 1/b$. Taking three times the derivative on both sides of (9) we obtain:

$$\sum_{n=0}^{k} \binom{k}{n} p^n(1-p)^{k-n}(n+1)x^n = (1-p+px)^k + x(1-p+px)^{k-1}kp \tag{10}$$

$$\sum_{n=0}^{k} \binom{k}{n} p^n(1-p)^{k-n}(n+1)nx^{n-1} = 2kp(1-p+px)^{k-1} + x(k-1)kp^2(1-p+px)^{k-2} \tag{11}$$

$$\sum_{n=0}^{k} \binom{k}{n} p^n(1-p)^{k-n}(n+1)n(n-1)x^{n-2} = 3k(k-1)p^2(1-p+px)^{k-2} + xk(k-1)(k-2)p^3 \times(1-p+px)^{k-3} \tag{12}$$

By setting $x = 1$, relations (11, 12) become:

$$\sum_{n=0}^{k}\binom{k}{n}p^n(1-p)^{k-n}(n+1)n = 2kp+k(k-1)p^2 \quad (13)$$

$$\sum_{n=0}^{k}\binom{k}{n}p^n(1-p)^{k-n}(n+1)n(n-1) = 3k(k-1)p^2+k(k-1)(k-2)p^3 \quad (14)$$

From relations (13–14) the second summation of (8) becomes:

$$\sum_{n=0}^{k}\binom{k}{n}p^n(1-p)^{k-n}n^2(n+1) = \sum_{n=0}^{k}\binom{k}{n}p^n(1-p)^{k-n}((n+1)n(n-1)+n(n+1)) = 2kp+4p^2k(k-1)+p^3k(k-1)(k-2) \quad (15)$$

Substituting (15) into (8) it is derived that the expected value of overflow accesses for the successful search of any record located in any bucket of the file as a function of the file expansion factor is given by:

$$S(\alpha) = \frac{1}{2X(\alpha)}\sum_{k=1}^{\infty}\left(2b+(k-1)\left(\frac{k-2}{b}+4\right)\right) \times\left(1-\left(1-\frac{1}{b}\right)^k\right)P(b+k,\alpha) \quad (16)$$

2.2 Unsuccessful search

For an unsuccessful search the probability of hitting a specific main bucket depends on the number of records it contained at initial loading, and not on the number of records at some point after loading.[8] Then the probability of hitting a main bucket allocated with m records at initial loading, is:

$$R_m m/W \quad \text{where } W = \sum_{j=1}^{b} jR_j \quad (17)$$

As in the case of the successful search, it is more probable to follow a longer chain than a shorter one, for an unsuccessful search as well. Assuming an overflow chain with n records, then the number of subintervals formed between the n records is $n+1$. The total number of subintervals in the overflow area of the bucket is $k+b(1-(1-1/b)^k)$. Therefore the expected value of subintervals per chain is $(k+b(1-(1-1/b)^k))/b(1-(1-1/b)^k)$. The probability of following a specific chain for an unsuccessful search is given by relation (18):

$$\frac{n+1}{\dfrac{(k+b(1-(1-1/b)^k))}{b(1-(1-1/b)^k)}} \quad (18)$$

The first subinterval of a chain is the one between a record of the main bucket and its first overflow record. To search for a key in this subinterval, only one additional access is needed to recognise its non-existence. In a similar manner, to search for a key in the lth subinterval (where $l < n$) l additional accesses are needed to perform the search unsuccessfully. However, n accesses are needed to perform an unsuccessful search for a key in the $(n+1)$th (last) subinterval. Hence we conclude that the number of accesses required to retrieve unsuccessfully the n records from the $n+1$ subintervals of the overflow chain is $1+2+\ldots+(n-1)+n+n = n(n+3)/2$. Since the key of the first record of every main bucket defines its associated track index, the $b+k$ records assigned to each main bucket form an equal number of subintervals. Therefore the expected value of overflow accesses required to perform the unsuccessful search in a specific overflow chain is:

$$n(n+3)/2(b+k) \quad (19)$$

The probability that this main bucket contains k overflow records at some point after loading is $P_m(b+k, \alpha)$. Taking into account that there are b chains and summing equations (3, 17–19) over j, k and n it is concluded that the expected value of overflow block accesses for an unsuccessful search is given by the relation:

$$U(\alpha) = \frac{b}{2W}\sum_{m=1}^{b} mR_m \sum_{k=1}^{\infty}\frac{P_m(b+k,\alpha)}{b+k} \times\sum_{n=1}^{k}\binom{k}{n}\left(\frac{1}{b}\right)^n\left(1-\frac{1}{b}\right)^{k-n}\frac{(n+1)b(1-(1-1/b)^k)}{k+b(1-(1-1/b)^k)}n(n+3) \quad (20)$$

Following a similar procedure as for the successful search and using equations (13–14) it is easily derived that:

$$\sum_{n=0}^{k}\binom{k}{n}\left(\frac{1}{b}\right)^n\left(1-\frac{1}{b}\right)^{k-n}n(n+1)(n+3) = \sum_{n=0}^{k}\binom{k}{n}\left(\frac{1}{b}\right)^n\left(1-\frac{1}{b}\right)^{k-n}((n+1)n(n-1)+4n(n+1)) = 8kp+7k(k-1)p^2+k(k-1)(k-2)p^3 \quad (21)$$

Substituting (21) into (20) we conclude that the expected value of overflow block accesses for an unsuccessful search in any bucket of the file as a function of the file expansion factor is:

$$U(\alpha) = \frac{1}{2W}\sum_{m=1}^{b} mR_m \sum_{k=1}^{\infty}\frac{k(1-(1-1/b)^k)}{(b+k)(k+b(1-(1-1/b)^k))} \times\left(8b+(k-1)\left(7+\frac{k-2}{b}\right)\right)P_m(b+k,\alpha) \quad (22)$$

3. NUMERICAL RESULTS AND DISCUSSION

Computations have been made using both the proposed model and the one studied by Larson for the case of a growing file. In a growing file either only insertions and no deletions occur or the deleted records are flagged. Numerical results have been obtained for various values of m for the cases of successful and unsuccessful searches (Figs 1(*a*), 2(*a*) and 1(*b*), 2(*b*) respectively. For simplification reasons the value of m was constant for all the main buckets and thus the value of W was equal to one.

In our mathematical analysis we have employed the binomial distribution for the lengths of the overflow chains. However, if the bucket size is very small, then the

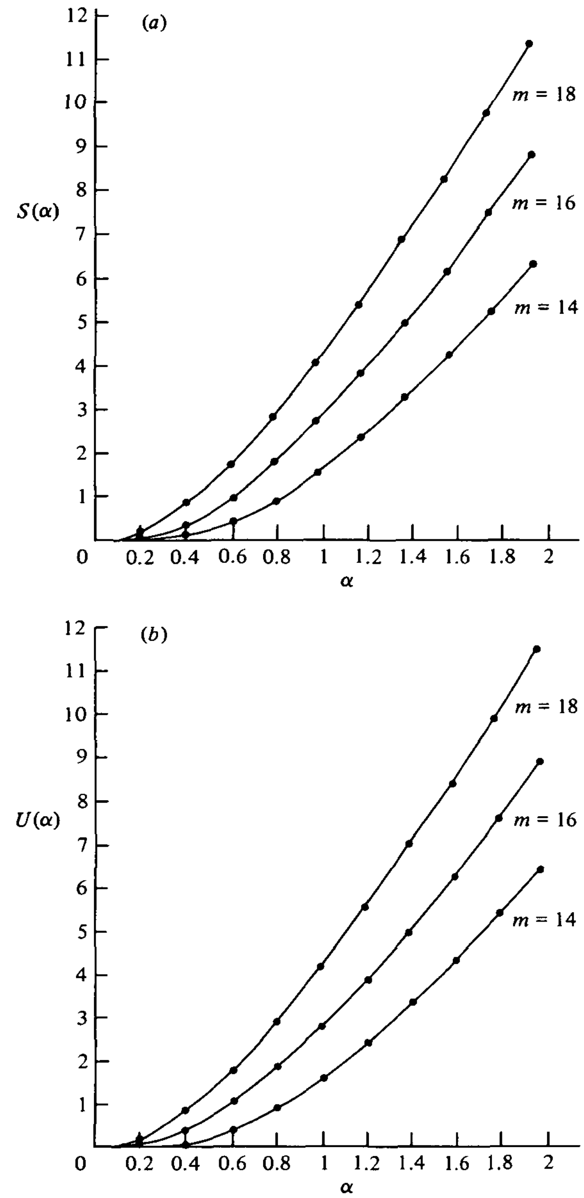

Figure 1. Expected number of overflow accesses for Larson's model in a growing file: (*a*) for successful searches; (*b*) for unsuccessful searches.

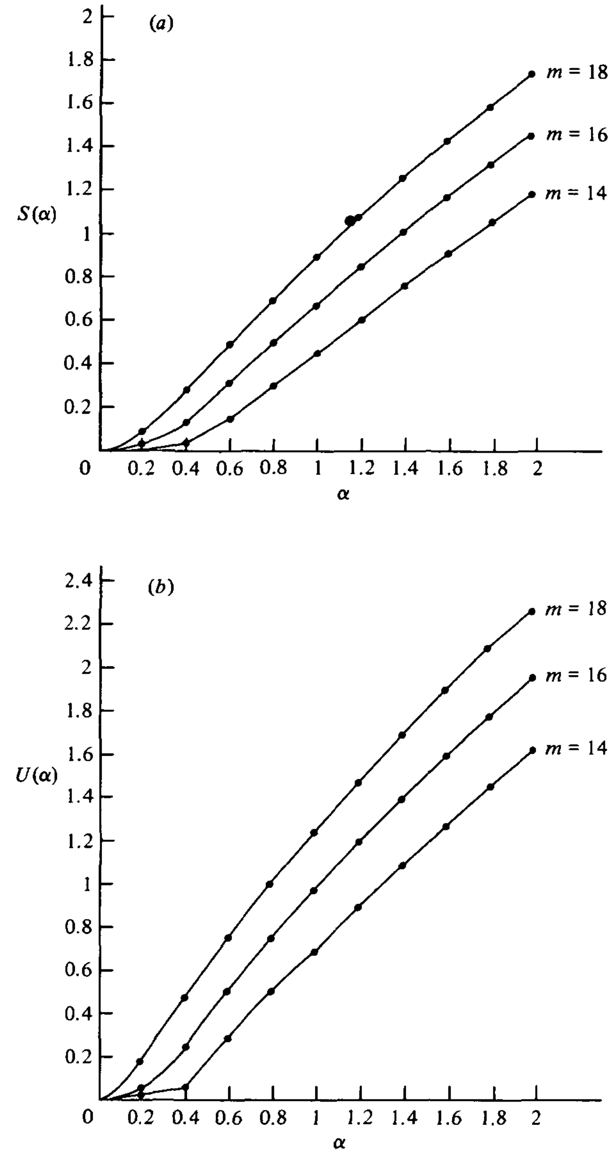

Figure 2. Expected number of overflow accesses for the proposed model in a growing file: (*a*) for successful searches; (*b*) for unsuccessful searches, using the approximate probability distribution function (1).

most appropriate distribution is the multinomial one. The analysis would have been further simplified, if it had been considered that the length of all the chains is equal to the mean value k/b. This simplification would have resulted in an optimistic evaluation, while by adopting the binomial distribution the approximation in the analysis is acceptable.

The assumption that every record of the overflow chain is retrieved with one additional access leads to a pessimistic performance evaluation. Since overflow blocks with capacity one are not often met in practice, and since the probability that two or more records residing in the same physical block is not zero, then the one record of an overflow chain can be retrieved in less than $l(l+1)/2$ accesses. The exact expected value of overflow accesses for this model can be determined by using non-homogeneous Markov chains.[11]

Referring to Fig. 1(*a*), the expected number of block accesses for a successful search increases linearly with a rate of $r_L = 6.11$ ($m = 16$) block accesses for a unity file expansion factor. In the proposed model (Fig. 2(*a*)) the expected number of block accesses increases with a rate of $r_P = 0.84$ ($m = 16$) block accesses for unity file expansion factor. This dramatic difference is due to the fact that in the proposed model every record of the main bucket is an index to a specific overflow chain. Therefore it can be considered that Larson's one overflow chain per bucket is split into b chains. Thus during the retrieval process fewer records are visited.

The rates of increase of the overflow accesses for unity file expansion factor can be used to estimate the appropriate point at which the file must be reorganised. In fact, if Larson's model needs to be reorganised at the point that the file expansion factor becomes equal to M, the life of our model is prolonged by $M(r_L/r_P)-M$ times.

Note at this point that in Section two it was assumed that all Q_is are fixed and equal to their mean value. In this way their stochastic nature was ignored. However, based on Batory's analysis,[1] a new asymptotic exact

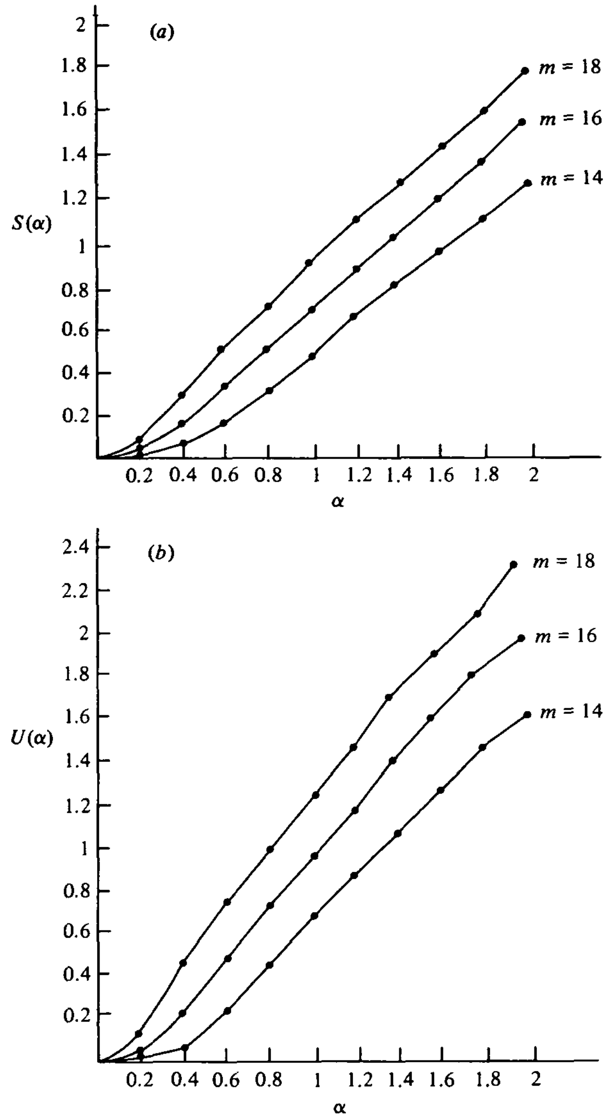

Figure 3. Expected number of overflow accesses for the proposed model in a growing file: (*a*) for successful searches; (*b*) for unsuccessful searches, using the exact asymptotic probability distribution function (23).

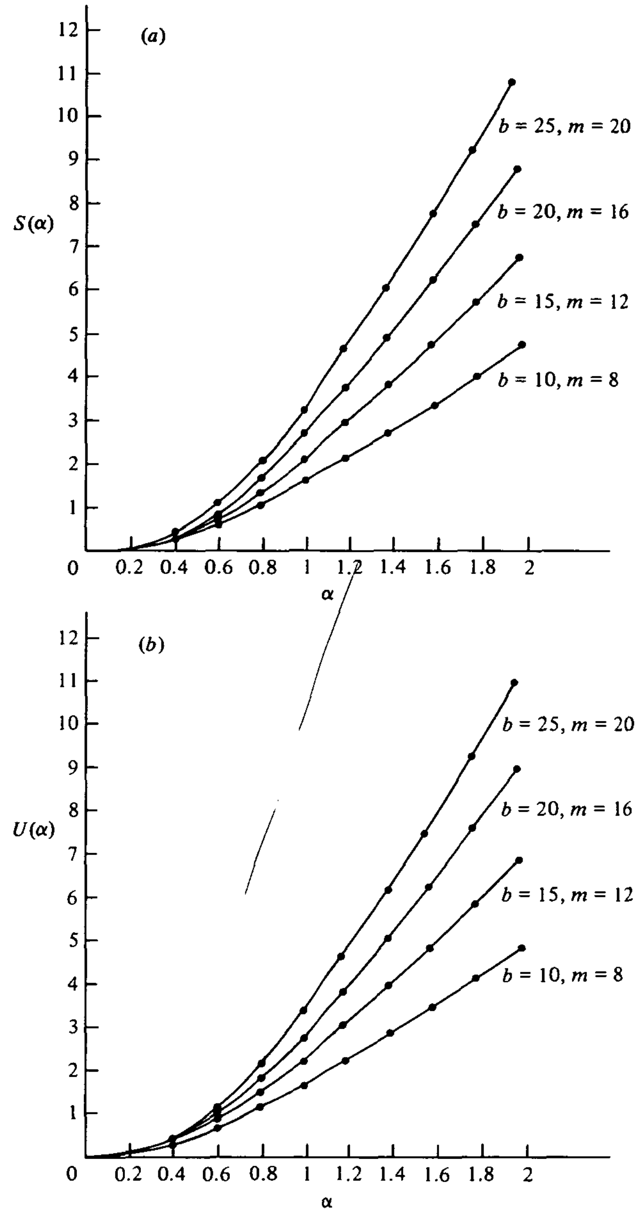

Figure 4. Expected number of overflow accesses for Larson's model for different values of *b* and *m* and a loading factor of 0.8: (*a*) for successful searches; (*b*) for unsuccessful searches.

probability distribution was derived in a recent work,[2] given by:

$$P(x-m,\, \alpha) = \binom{x-1}{x-m} \frac{\alpha^{x-m}}{(1+\alpha)^{x}} \tag{23}$$

By employing this distribution function in our model, the expected values for successful and unsuccessful searches for different values of m are given in Figs 3(*a*) and 3(*b*) respectively. For this case the expected number of block accesses increases now with a rate of $r_p = 0.88$ ($m = 16$) block accesses for unity file expansion factor. Therefore with the exact probability distribution function the expected number of block accesses is increased by 5% compared to the results obtained by employing the approximate probability distribution function.

One of the characteristics of our model is that the number of overflow accesses depends only on the loading factor (lf), while in Larson's model this number is dependent on lf and b. Computations have been carried out for a loading factor, $lf = m/b = 0.8$, and for the cases of $b = 10$, $m = 8$; $b = 15$, $m = 12$; $b = 20$, $m = 16$ and $b = 25$, $m = 20$. For the proposed model, the expected number of overflow accesses for successful search as a function of the file expansion factor is represented by the second curve of Fig. 2(*a*). For Larson's model a family of curves is obtained, as is shown in Fig. 4. The different behaviour of the new model can be explained as follows. Consider the point that the file expansion factor is equal to i. Deterministically the number of overflow records, k, at this point is $(i+1)\,m-b$. Therefore the mean value of overflow accesses, for Larson's model, is given by the following formula:

$$\frac{k(k+1)}{2(i+1)\,m}\;\frac{[(i+1)\,lf-1]\,[b(i+1)\,lf-b+1]}{2(i+1)\,lf} \tag{24}$$

In our model there are b overflow chains, each one having an expected length of $c = k/b$, and therefore the mean value of overflow accesses is given by the relation (25):

$$\frac{bc(c+1)}{2(1+1)m}\,\frac{(i+1)lf-1}{2} \tag{25}$$

Similar conclusions are derived for the case of unsuccessful search. Hence the task of predicting the performance of the proposed ISAM file for a given loading factor can be easily determined from predefined curves.

Our model presents an additional advantage regarding the insertion of new records into a bucket. In practice, in ISAM files a continuous movement of records from the main bucket to the overflow area occurs, in order to keep the main bucket full. However, in the proposed model, since each record of the main bucket is an index for its associated overflow chain, no local reorganisation of the main bucket is required. This merit results in faster file-handling software.

Acknowledgements

The authors would like to thank the anonymous referee and Professor Stavros Christodoulakis for their valuable comments.

REFERENCES

1. D. Batory, Optimal file designs and reorganization points. *ACM Transactions on Database Systems*, **7** (1), 60–81 (1982).
2. S. Christodoulakis, Y. Manolopoulos and P. A. Larson, File organisations for variable length records. *ACM Transactions on Database Systems*, in review (1987).
3. R. B. Cooper and M. K. Solomon, The average time until bucket overflow. *ACM Transactions on Database Systems* **9** (3), 392–408 (1984).
4. W. Feller, *Introduction to Probability Theory and its Applications*, vol. 1, 3rd edn. Wiley, Chichester (1968).
5. D. P. Heyman, Mathematical models of database degradation. *ACM Transactions on Database Systems* **7** (4), 615–631 (1982).
6. D. G. Keehn and J. O. Lacy, VSAM data set design parameters. *IBM Systems Journal* **13** (3), 186–212 (1974).
7. D. Knuth, *The Art of Computer Programming: Fundamental Algorithms*, vol. 1, 2nd edn. Addison-Wesley, New York (1973).
8. P. A. Larson, Analysis of index sequential files with overflow chaining. *ACM Transactions on Database Systems* **6**, (4), 671–680 (1981).
9. P. A. Larson, Performance analysis of a single-file version of linear hashing. *The Computer Journal* **28**, (3), 319–329 (1985).
10. Y. Manolopoulos, Batched search of index sequential files. *Information Processing Letters*, **22**, 267–272 (1986).
11. Y. Manolopoulos, Database Performance Evaluation when Searching for Simple and Batched Queries. *Ph.D. dissertation* (in Greek), Faculty of Electrical Engineering, University of Thessaloniki (1986).
12. T. J. Teorey and J. P. Fry, *Design of Database Structures*. Prentice-Hall, Englewood Cliffs, N.J. (1983).
13. J. P. Tremblay and P. G. Sorenson, *An Introduction to Data Structures with Applications*, 2nd edn. McGraw-Hill, London (1984).
14. G. Weilderhold *Database Design*, 2nd edn. McGraw-Hill, London (1983).