A uniform model for the storage utilization of B-tree-like structures

Donatella Merlini, Renzo Sprugnoli, M. Cecilia Verri
Dipartimento di Sistemi e Informatica
Via Lombroso 6/17- Firenze
e-mail: resp@dsi2.ing.unifi.it

Abstract

By using the method of Riordan arrays we find the asymptotic value of the expected storage utilization in a model of uniform B-tree-like structures proposed by Gupta and Srinivasan, thus solving the occupancy problem for that model.

Keywords: analysis of algorithms, B-trees, Riordan arrays, storage utilization.

1 Introduction

B-trees are one of the most frequently used structures for storing information in a two level storage. Since their introduction in the early '70, they have been widely studied together with their main memory companion, the 2-3-trees, and therefore a vast literature exists, in whichtheir performance is analyzed. Just to give some examples, we quote here the papers by Baeza-Yates [1], Chu and Knott [3], Quitzow and Klopprogge [9] and Wright [14], which all refer to a distribution in which key permutations are all equally likely. The problem is quite hard, and therefore several approximate models have been proposed, in order to simplify the analysis, especially concerning their "occupancy problem", i.e. how much space is necessary to store a given number of keys. If we have r keys, stored in n nodes, each containing a maximum of $2d$ keys (d is the *order* of the tree), then the ratio $r/(2dn)$ is called the *load factor* for the B-tree, and is the quantity of main interest.

The approximate model introduced by Leung [8] in 1984 was rediscussed by Gupta and Srinivasan [5], who point out that the model refers to a uniform distribution of B-trees, i.e., a distribution in which all possible configurations of B-trees with the same number of keys are equally likely. To be more precise, if f is the minumun storage utilization of each node, then each node can contain at least $2df$ keys and, obviously, at most $2d$ keys; Gupta and Srinivasan observe that the problem of finding the number of different B-tree configurations having an order d, n nodes and r keys is equivalent to the restricted occupancy problem when n different cells can contain no more than v of p like objects (see Riordan [10] p. 104), where $p = r - 2dnf$ and $v = (1 - f)2d$.

Unfortunately, this statement is not precise and Gupta and Srinivasan's model cannot be considered to be a true model for B-trees. For example, if we have 12 keys, then we cannot arrange them in a B-tree having only 3 4-key nodes ($d = 2$), even though this situation ($r = 12, n = 3$) is taken into account by the model, because the corresponding restricted occupancy problem can be solved by arranging $p = 6$ objects in three cells, each of which can contain no more than $v = 2$ objects. In general, we can point out that Gupta and Srinivasan's model ignores the peculiar structure of B-trees, as well as the fact that they are balanced and the number of any node's siblings is strictly related to the number of the keys in the same node. Actually, the model seems to refer to non-balanced trees, whose nodes are similar to the ones in a B-tree, but whose pointer structure is much more free, in the sense that a random number of subtrees can originate from a node. From here on, we call the structures defined by Gupta and Srinivasan's model, *(B)-trees*, to emphasize the fact that they are a loose model of B-trees and similar structures. They could also be used as a model for dynamic hashing (see Larson [7]), if the keys were evenly distributed between the new nodes whenever we split a node; in this case, the nodes are simply key containers and are not related to each other.

It makes sense to reconsider Gupta and Srinivasan's model from this point of view and in fact we obtain more precise results on storage utilization. By the way, from a mathematical point of view, these results conclusively show that t he model cannot be considered a realistic B-tree model.

Let us now come to the technical part of Gupta and Srinivasan's paper. They show that, if $N_p(n, v)$ represents the number of B-trees with p keys distributed over n nodes such that each node can have a maximum of v vacancies, then:

$$N_p(n, v) = \sum_{k=0}^{n} (-1)^k \binom{n}{k} \binom{p - kv - k + n - 1}{p - kv - k}. \tag{1.1}$$

In this model, the number n of nodes varies between $n_{min} = r/(2d)$ and $n_{max} = r/(2df)$ and the expected storage utilization is:

$$E(u) = \frac{r \sum_{n=n_{min}}^{n_{max}} N_p(n, v)}{2d \sum_{n=n_{min}}^{n_{max}} n N_p(n, v)} = \frac{r}{2d} E\left(\frac{1}{n}\right).$$

This definition of $E(1/n)$ is slightly different from Gupta and Srinivasan's but it is more appropriate from a probabilistic point of view. The sums can be extended from 0 to ∞ because by (1.1) the value of $N_p(n, v)$ is zero when n is outside the range $n_{min}..n_{max}$.

Gupta and Srinivasan give some examples and obtain that the number of nodes is not uniformly distributed, which is a positive result. However, they conclude that "the problem of deriving an expression for the average value of n has not been resolved. Given the complex expression for $N_p(n, v)$, obtaining a closed expression for $E(1/n)$ seems unlikely."

In the present paper, we want to show that it is possible to find the generating function for the sums $\sum_n N_p(n, v)$ and $\sum_n n N_p(n, v)$. This allows us to give either approximate or asymptotic estimates for the expected value $E(u)$, and thus solve the storage utilization problem for this model. To this purpose, we use the method of *Riordan arrays* which will be introduced in Section 2. Riordan arrays constitute a flexible algebraic approach to the solution of combinatorial sums, to find either their closed form or their asymptotic estimate, and seem to have many applications to combinatorial analysis and to analysis of algorithms.

2 Riordan Arrays

The technique we use to solve the occupancy problem relies on the concept of *Riordan array* (see Shapiro et al. [11] and Sprugnoli [12]). A Riordan array $\{a_{p,k} | p, k \in N, k \leq p\}$ is an infinite, low triangular array defined by means of two analytic functions $(b(t), h(t))$, in the sense that the generic element $a_{p,k}$ is given by:

$$a_{p,k} = [t^p]b(t)(th(t))^k,$$

$[t^p]$ being the "coefficient of" operator. In other words, $b(t)(th(t))^k, k = 0, 1, \dots,$ is the generating function of the k^{th} column of the Riordan array. Riordan arrays allow us to solve combinatorial sums like $\sum_k a_{p,k} g_k$ in terms of generating functions according to the following rule:

$$\mathcal{G}\left\{\sum_{k=0}^{\infty} a_{p,k} g_k\right\} = b(t)g(th(t)), \tag{2.2}$$

where the symbol $\mathcal{G}$ denotes the generating function operator, i.e.:

$$g(t) = \mathcal{G}\{g_k\}_{k\in N} = \mathcal{G}\{g_k\} = \sum_{k=0}^{\infty} g_k t^k$$

is the generating function of the sequence $\{g_k\}_{k\in N}$ (see Bender and Williamson [2]). In particular, the array's row sums and weighted row sums, in which $g_k = 1$ and $g_k = k$, have the following generating functions:

$$\mathcal{G}\left\{\sum_k a_{p,k}\right\} = \frac{b(t)}{1 - th(t)}, \qquad \mathcal{G}\left\{\sum_k k a_{p,k}\right\} = \frac{tb(t)h(t)}{(1 - th(t))^2}.$$

We wish to point out that in formula (1.1),

$$a_{p,k} = \binom{p + n - 1 - k(v + 1)}{p - k(v + 1)}$$

is the Riordan array $D = (1/(1 - t)^n, t^v)$ and by a simple application of rule (2.2), with $g(t) = (1 - t)^n$ (the generating function of binomial coefficients with alternating signs), we obtain:

$$N_p(n, v) = \sum_k \binom{p + n - 1 - k(v + 1)}{p - k(v + 1)} \binom{n}{k} (-1)^k = [t^{p+n-1}] \frac{t^{n-1}}{(1 - t)^n} \left(1 - t^{v+1}\right)^n =$$

$$= [t^p] \left(\frac{1 - t^{v+1}}{1 - t}\right)^n = [t^{r-2dnf}] \left(1 + t + \dots + t^{2(1-f)d}\right)^n = [t^r] \left(t^{2fd} \left(1 + \dots + t^{2(1-f)d}\right)\right)^n.$$

This proves that, when $2df$ and $2(1 - f)d$ are integer numbers, then $N_p(n, v)$ is the generic element of the Riordan array defined by $b(t) = 1$ and $h(t) = t^{2df-1} \left(1 + t + \dots + t^{2(1-f)d}\right)$. This immediately gives:

$$S = \sum_n N_p(n, v) = [t^r] \frac{1}{1 - t^{2df} \left(1 + t + \dots + t^{2(1-f)d}\right)},$$

$$W = \sum_n n N_p(n,v) = [t^r] \frac{t^{2df}\left(1+t+\ldots+t^{2(1-f)d}\right)}{\left(1-t^{2df}\left(1+t+\ldots+t^{2(1-f)d}\right)\right)^2}.$$

The most important cases are:

i) $f = 1/2$. This corresponds to (B)-trees whose analogue is the standard definition of B-trees.

ii) $f = 2/3$. This corresponds to (B)-trees with left [right] rotations. In the B-tree analogue, when a node is full, its keys are "rotated" into the left [right] sibling. Node splitting is performed only when the two nodes are full.

iii) $f = 3/4$. This corresponds to (B)-trees with left and right rotations. In the B-trees analogue, when a node is full, the keys are rotated into the left or right sibling, according to whichever has room. Node splitting is performed only when the three nodes are full.

In any case, we can always choose d in such a way that both $2df$ and $2(1-f)d$ are integer. The expected storage utilization is given by:

$$E(u) = \frac{rS}{2dW}.$$

If we now set:

$$P(t) = 1 - t^{2df}\left(1+t+\ldots+t^{2d(1-f)}\right) = 1 - t^{2df}\frac{1-t^{2d(1-f)+1}}{1-t},$$

$$Q(t) = 1+t+\ldots+t^{2d(1-f)},$$

it is obvious that the asymptotic values of S and W are connected to the zeroes of $P(t)$ of minimal modulus (see Flajolet and Odlyzko [4] and Wilf [13] for a detailed explanation of singularity analysis for generating functions). By applying the results about the location of the zeros of polynomials (see for example Henrici [6], Th. 6.4b and Cor. 6.4k), we find that the zeroes of $P(t)$ lie in the annulus bounded by the two concentric circles of radii $1/2$ and 2, centered at the origin. We observe that, by adding the root $t = 1$ to $P(t)$, we get the equivalent equation $(1-t)P(t) = 1 - t - t^{2df} + t^{2d+1} = 0$. By applying Descartes' rule to this polynomial (see Henrici [6]), we find that if the polynomial has positive real roots, then it must have exactly two, and if a real negative root exists, it must be unique. Moreover, the following statements can be easily proved:

- $P(t) > 0$ for $-1 < t \leq 0$ $\quad$ ($P(-1)$ may be 0 or 1.)
- $P(0) = 1$ and $P(1) < 0$.

This implies that only two real roots exist for $P(t) = 0$, s and $\bar{s}$, with $1/2 < s < 1$ and $-2 < \bar{s} \leq -1$ and multiplicity 1. The other $2d-2$ roots are all complex. Since the coefficients of the Taylor expansion of $P(t)^{-1}$ around $t = 0$ are all positive (i.e., our $N_p(n,v)$'s), s should be the root of minimal modulus, and thus we have:

$$\frac{1}{P(t)} \sim \frac{A}{1-t/s}, \qquad S \sim \frac{A}{s^r}$$

$$\frac{Q(t)}{P(t)^2} \sim \frac{B}{(1-t/s)^2} + \frac{C}{1-t/s}, \qquad W \sim \frac{B(r-2df+1)}{s^{r-2df}} + \frac{C}{s^{r-2df}}$$

for suitable constants A, B and C, which are easily computed in terms of s:

$$A = -\frac{1}{sP'(s)}, \qquad B = \frac{Q(s)}{(sP'(s))^2}, \qquad C = \frac{Q(s)P''(s) - Q'(s)P'(s)}{sP'(s)^2}.$$

A's and B's values are the most important ones, and can be described in further detail by observing that:

$$Q(s) = 1 + s + s^2 + \ldots + s^{2d(1-f)} = \frac{1 - s^{2d(1-f)+1}}{1-s},$$

$$P'(s) = -2dfs^{2df-1}\frac{1-s^{2d(1-f)+1}}{1-s} - s^{2df}\frac{1 - s^{2d(1-f)+1} - 2d(1-f)s^{2d(1-f)}(1-s)}{(1-s)^2} =$$

$$= -2dfs^{2df-1}\frac{1-s^{2d(1-f)+1}}{1-s} - \frac{1-(2d(1-f)+1)\,s^{2d}}{1-s}.$$

Here we use the identity $s^{2df}\left(1 - s^{2d(1-f)+1}\right) = 1-s$, because s is a root of $P(t)$. We can now compute the expected value of storage utilization as follows:

$$E(u) \sim \frac{rA}{s^{2df}2d\,(B(r-2df+1)+C)} \sim \frac{A}{2ds^{2df}B} = \frac{-P'(s)}{2ds^{2df-1}Q(s)}.$$

We set $r \to \infty$, i.e., we assumed that our (B)-trees contain a large number of keys. By inserting the above values of $P'(s)$ and $Q(s)$ in this expression, we have:

$$E(u) \sim f + \frac{s\left(1-(2d(1-f)+1)\,s^{2d}\right)}{2d(1-s)},$$

which is our most significant result. It gives us the expected value of storage utilization in terms of s, the unique real solution of $P(t) = 0$ such that $1/2 < s < 1$. This result gives us some further information: by construction, we know that $E(u) \geq f$, and this means that $1 - (2d(1-f)+1)s^{2d} \geq 0$. This provides us with an upper limit for s:

$$s \leq \frac{1}{\sqrt[2d]{2d\,(1-f)+1}}.$$

If we set $e^h = 2d(1-f)+1$ or $h = \ln(2d(1-f)+1)$ in the relation $(1+h/(2d))^{2d} \sim e^h$, we obtain the approximation $\sqrt[2d]{2d(1-f)+1} \sim 1 + \ln(2d(1-f)+1)/2d$, or:

$$s \leq 1 - \frac{\ln(2d\,(1-f)+1)}{2d}.$$

Actually, this is a good approximation of s, which can also be used to show that $E(u) \to f$ when $d \to \infty$, and obviously remains $d = o(r)$. Fortunately, when d is in a practical range (for example, when the nodes contain from 10 to 400 keys), the expected storage utilization is much larger than f, as Table 1 shows. However it can be noted that, in this model, storage utilization decreases as the dimension of the nodes increases. These results agree with the intuitive idea that, if we have r keys ($r \to \infty$) which have to be randomly distributed in n cells containing at least $2df$ keys and at most $2d$ keys, the more d increases, the more empty space is left in the cells.

d	$f=1/2$		$f=2/3$		$f=3/4$	
	s	$E(u)$	s	$E(u)$	s	$E(u)$
6	0.796544	67.72%	0.849077	80.64%	0.875395	86.12%
12	0.859757	66.71%	0.894105	80.28%	0.910745	85.95%
24	0.908698	65.50%	0.930175	79.80%	0.940164	85.72%
48	0.943159	64.21%	0.956165	79.25%	0.961960	85.43%
96	0.965760	62.92%	0.973493	78.64%	0.976783	85.10%
192	0.979948	61.70%	0.984418	78.01%	0.986260	84.76%

Table 1: $E(u)$ =Approximate storage utilization

3 The $f = 0.5$ case

When $f = 0.5$, that is, when the B-tree analogues have their standard formulation, the formulas involved simplify and therefore allow us to be more precise. In particular, the relation $P(t) = 0$ can be written in the form $t^d(1 - t^{d+1}) = 1 - t$ or $t(1 - t^{2d}) = 1 - t^d$, which simplifies to $t(1 + t^d) = 1$. It is worth noting that in the former form, we add the root $t = 1$ to $P(t)$, while in the latter, we eliminate the d^{th} roots of unity. Therefore, $1 - t - t^{d+1}$ has the same roots as $P(t)$, except for the d^{th} roots of unity. The latter, however, are not significant in the asymptotic evaluation of S and W. The real root s $(1/2 < s < 1)$ satisfies the identity $s^{d+1} = 1 - s$, and this can be used to simplify all our expressions. We can also easily prove a stronger limitation for s, as follows:

$$1 - \frac{\ln(d+1)}{d+1} < s < 1 - \frac{\ln(d+1)}{d+1} + \frac{\ln(\ln(d+1))}{d+1}.$$

In fact, if we denote the two values by x and y, i.e. $x < s < y$, we can show that $1-x-x^{d+1} > 0$ and $1 - y - y^{d+1} < 0$, which means that x and y separate the root s. In fact, by writing m for $d + 1$, we find:

$$1 - \left(1 - \frac{\ln m}{m}\right) - \left(1 - \frac{\ln m}{m}\right)^m = \frac{\ln m}{m} - \exp\left(m \ln\left(1 - \frac{\ln m}{m}\right)\right) =$$

$$= \frac{\ln m}{m} - \exp\left(-\ln m - \frac{(\ln m)^2}{2m} - \frac{(\ln m)^3}{3m^2} + \ldots\right) =$$

$$= \frac{\ln m}{m} - \frac{1}{m} + \frac{(\ln m)^2}{2m^2} + O\left(\frac{(\ln m)^4}{m^3}\right),$$

which is positive for $m \geq 3$ or $d \geq 2$. In a similar way, we can show that $1 - y - y^{d+1} < 0$.

By applying the rule $s^{d+1} = 1 - s$, we find the values:

$$Q(s) = \frac{s}{1-s}, \qquad P'(s) = \frac{(2s-1)(d - ds + 1)}{s^2(1-s)}$$

and therefore:

$$E(u) \sim \frac{(2s-1)(d-ds+1)}{2ds(1-s)} = \left(1-\frac{1}{2s}\right)\left(1+\frac{1}{d(1-s)}\right).$$

By means of this formula, we deduce the asymptotic behaviour $E(u) \sim 1/2 + 1/2\ln(d+1)$, which explains the slow convergence towards 1/2 of the expected storage utilization.

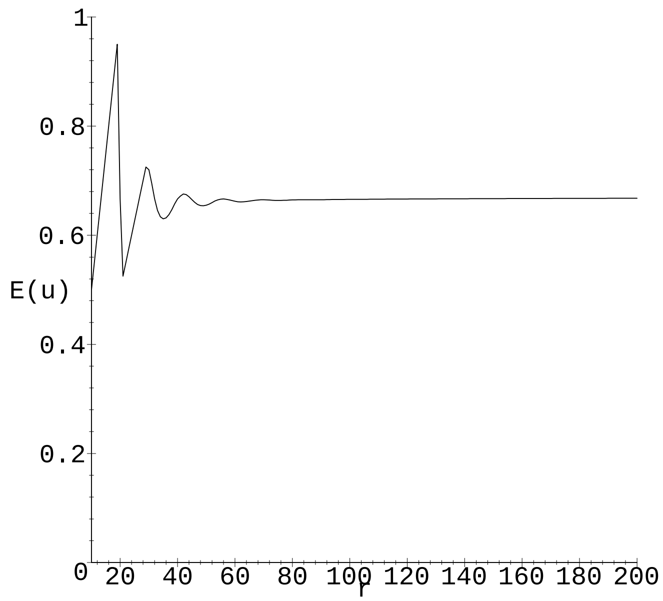

Figure 1: $E(u)$ for $d = 10$ and $f = 0.5$

We conclude by pointing out that we only examined the root s of minimal modulus. As previously mentioned, we also have $2d-2$ complex conjugate roots, some of which have modulus less than 1. By applying Euclid's algorithm to $P(t)$ and $P'(t)$, we can show that all roots have multiplicity one. These roots produce oscillations in the behaviour of $E(u)$, which tend to disappear as r increases. For example, when $d = 10$, the complex roots having minimal modulus are:

$$s_1 \approx 0.831630 - 0.415552i \qquad s_2 \approx 0.831630 + 0.415552i \qquad |s_1| = |s_2| \approx 0.929673$$

and the minimal modulus' real root is:

$$s_3 \approx 0.844398.$$

In this case, S and W are more accurately described by the formulas:

$$S \sim \sum_{i=1}^{3} \frac{A_i}{s_i^r},$$

$$W \sim \sum_{i=1}^{3} \frac{B_i(r-2df+1)}{s_i^{r-2df}} + \frac{C_i}{s_i^{r-2df}},$$

in which the constants A_i, B_i, C_i correspond to A, B, C evaluated in $s = s_i, i = 1, 2, 3$, respectively. Figure 1 shows the oscillations in the behaviour of $E(u)$ for $d = 10$.

Acknowledgments

We thank prof. R. Pinzani for having brought Gupta and Srinivasan's paper to our attention and prof. Ph. Flajolet for useful discussions on the adequacy of Gupta and Srinivasan's model as a B-tree model.

References

[1] R. A. Baeza-Yates. Expected behaviour of B^+-trees under random insertions. *Acta Informatica*, 26:439–471, 1989.

[2] E. A. Bender and S. G. Williamson. *Foundations of applied combinatorics.* Addison Wesley, 1991.

[3] J.-H. Chu and G. Knott. An analysis of B-trees and their variants. *Information Systems*, 14(5):359–370, 1989.

[4] P. Flajolet and A. M. Odlyzko. Singularity analysis of generating functions. *Siam Journal Discrete Mathematics*, 3(2):216–240, 1990.

[5] G.K. Gupta and B. Srinivasan. Approximate storage utilization of B-trees. *Information Processing Letters*, 22:243–246, 1986.

[6] P. Henrici. *Applied and Computational Complex Analysis. Vol. I.* Wiley, New York, 1988.

[7] P. Larson. Dynamic hashing. *Bit*, 18:184–201, 1978.

[8] C. H. C. Leung. Approximate utilization of B-trees: a simple derivation and generalisations. *Information Processing Letters*, 19:199–201, 1984.

[9] K. H. Quitzow and M. R. Klopprogge. Space utilization and access path length in B-trees. *Information Systems*, 5:7–16, 1980.

[10] J. Riordan. *An introduction to combinatorial analysis.* Wiley, New York, 1958.

[11] L. V. Shapiro, S. Getu, W.-J. Woan, and L. Woodson. The Riordan group. *Discrete Applied Matematics*, 34:229–239, 1991.

[12] R. Sprugnoli. Riordan arrays and combinatorial sums. *Discrete Mathematics*, 132:267–290, 1994.

[13] H. S. Wilf. *Generatingfunctionology.* Academic Press, 1990.

[14] W. E. Wright. Some average performance measures for the B-tree. *Acta Informatica*, 21:541–557, 1985.