Discrete Mathematics and Theoretical Computer Science DMTCS vol. **12:2**, 2010, 307–332

Distributional Analysis of the Parking Problem and Robin Hood Linear Probing Hashing with Buckets

Alfredo Viola[†]

Pedeciba Informática, Montevideo, Uruguay.

received Sep 1, 2009, *revised Apr 5, 2010*, *accepted Apr 15, 2010*.

This paper presents the first distributional analysis of both, a parking problem and a linear probing hashing scheme with buckets of size b. The exact distribution of the cost of successful searches for a $b\alpha$-full table is obtained, and moments and asymptotic results are derived. With the use of the Poisson transform distributional results are also obtained for tables of size m and n elements. A key element in the analysis is the use of a new family of numbers, called Tuba Numbers, that satisfies a recurrence resembling that of the Bernoulli numbers. These numbers may prove helpful in studying recurrences involving truncated generating functions, as well as in other problems related with buckets.

Keywords: Distributional Analysis, Hashing, Linear Probing, Buckets

1 Motivation and previous results

Throughout this paper we consider hash tables that have m locations (m is called the "length" of the table) each of them containing at most $b \geq 1$ keys, and we let n (with $0 \leq n \leq bm$) denote the total number of keys (the "size") in the table. The ratio $\alpha = n/bm$ is called the "load factor" of the table. Clearly, the number of tables (the number of hash sequences) with length m and size n is m^n ,

The simplest collision resolution scheme for open addressing hash tables with hash function $h(x)$ is *linear probing* [19, 29, 45], which uses the cyclic probe sequence $h(K), h(K)+1, \ldots m-1, 0, 1, \ldots, h(K)-1$, assuming the table slots are numbered from 0 to $m-1$. Linear probing works reasonably well for tables that are not too full, but as the load factor increases, its performance deteriorates rapidly. Its main application is to retrieve information in secondary storage devices when the load factor is not too high, as first proposed by Peterson [41]. One reason for the use of linear probing is that it preserves locality of reference between successive probes, thus avoiding long seeks [35].

[†]This work of was supported in part by proyecto CSIC fondos 2002-2004 and 2004-2006, from the Universidad de la República. Part of the work was done while the author was at LIPN - CNRS UMR 7030. Université de Paris-Nord. 93430 Villetaneuse, France.

For each element x that gets placed at some location y, the circular distance between y and $h(x)$ (that is, $y - h(x)$ if $h(x) \leq y$, and $m + h(x) - y$ otherwise) is called its *displacement*. Displacement is both a measure of the cost of inserting x and of the cost of searching x in the table. *Total displacement* corresponding to a sequence of hashed values is the sum of the individual displacements of elements. and it determines the *construction cost* of the table.

Linear probing hashing has been the object of intense study; see the table on results and the bibliography in [19, pp. 51-54]. The first published analysis of linear probing was done by Konheim and Weiss [34]. In addition, there is also special value for these problems since the first analysis of algorithms ever performed by D. Knuth [26] was that of linear probing hashing. As Knuth indicates in many of his writings, the problem has had a strong influence on his scientific carreer. Moreover, the construction cost to fill a linear probing hash table connects to a wealth of interesting combinatorial and analytic problems. More specifically, the Airy distribution that surfaces as a limit law in this construction cost is also present in random trees (inversions and path length), random graphs (the complexity or excess parameter), and in random walks (area) [33, 15].

Operating primarily in the context of double hashing, several authors [5, 2, 20] observed that a collision could be resolved in favor of *any* of the keys involved, and used this additional degree of freedom to decrease the expected search time in the table. We obtain the standard scheme by letting the incoming key probe its next location. So, we may see this standard policy as a *first-come-first-served* (FCFS) heuristic. Later Celis, Larson and Munro [8, 9] were the first to observe that collisions could be resolved having *variance reduction* as a goal. They defined the Robin Hood heuristic, in which each collision occurring on each insertion is resolved in favor of the key that is farthest away from its home location. Later, Poblete and Munro [43] defined the *last-come-first-served* (LCFS) heuristic, where collisions are resolved in favor of the incoming key, and others are moved ahead one position in their probe sequences. These strategies do not look ahead in the probe sequence, since the decision is made before any of the keys probes its next location. As a consequence, they do not improve the average search cost.

It is shown in [7] that the Robin Hood linear probing algorithm minimizes the variance of all linear probing algorithms that do not look ahead. This variance, for a full table, is $\Theta(m)$, instead of the $\Theta(m^{3/2})$ of the standard algorithm. They derived the following expressions for the variance of $C_{m,n}$, the successful search time

$$\begin{aligned}
\mathbf{Var}[C_{m,n}] &= \frac{1}{2}Q_1(m,n-1) - \frac{1}{4}Q_0(m,n-1)^2 - \frac{1}{6}Q_0(m,n-1) + \frac{1}{6}\frac{n-1}{m} - \frac{1}{12}, \\
\mathbf{Var}[C_{m,\alpha m}] &= \frac{1}{4(1-\alpha)^2} - \frac{1}{6(1-\alpha)} - \frac{1}{12} + \frac{\alpha}{6} - \frac{1}{6m} - \frac{1+2\alpha}{3(1-\alpha^4)m} + O\left(\frac{1}{m^2}\right), \\
\mathbf{Var}[C_{m,m}] &= \frac{4-\pi}{8}m + \frac{1}{9} - \frac{\pi}{48} + \frac{1}{135}\sqrt{\frac{2\pi}{m}} + O\left(\frac{1}{m^2}\right).
\end{aligned} \tag{1}$$

Moreover, in [25] and [48], a distributional analysis for the FCFS, LCFS and Robin Hood heuristic is presented. More specifically, for the Robin Hood heuristic, they obtain

$$
\begin{aligned}
Pr\{C_{m,n}=r\} &= \sum_{i=r}^{n+1}\binom{n+1}{i}\frac{(m-n-1+i)}{(n+1)m^i}\sum_{k=0}^{i+1-r}(-1)^{i+1-r-k}\binom{i+1}{r+k}k^i,\\
Pr\{C_{m,\alpha m}=r\} &= (1-\alpha)\sum_{i\geq r}\frac{\alpha^{i-1}}{i!}\sum_{k=1}^{i+1-r}(-1)^{i+1-k-r}\binom{i+1}{r+k}k^i\\
-\frac{1}{2m}&\left(\sum_{i\geq r}\frac{\alpha^i(i-1)(i-2-i\alpha)}{i!}\sum_{k=1}^{i+1-r}(-1)^{i+1-k-r}\binom{i+1}{r+k}k^i\right)+O\left(\frac{1}{m^2}\right),\\
C_\alpha(z) &= \lim_{m\to\infty}\sum_{r=0}^{\infty}Pr\{C_{m,\alpha m}=r\}z^r = z\frac{1-\alpha}{\alpha}\frac{e^{z\alpha}-e^{\alpha}}{ze^{\alpha}-e^{z\alpha}},
\end{aligned}
$$

where $C_\alpha(z)$ is the probability generating function of the successful search time when $m,n\to\infty$ and $m/n=\alpha$, $0\leq\alpha<1$.

These results consider a hash table with buckets of size 1. However, very little is known when we have tables with buckets of size b. In [4], Blake and Konheim studied the asymptotic behavior of the expected cost of successful searches as the number of elements and buckets tend to infinity with their ratio remaining constant. Mendelson [36] derived exact formulae for the same expected cost, but only solved them numerically. These papers consider the FCFS heuristic. In [49] the first exact analysis of a linear probing hashing scheme with buckets of size b is presented. In that paper, they find the expected value and the asymptotic behavior of the average cost of successful searches when the Robin Hood heuristic is used. One of their main methodological contributions is the introduction of a new sequence of numbers $T_{k,d,b}$ for $0\leq d<b$ (that we call *Tuba Numbers* [(i)]), and is one of the key components of the analysis presented in this paper. This is sequence *EIS* **A124453** in Neil Sloane's *Encyclopedia of Integer Sequences*.

In this paper we complete the work presented in [49], and find the distribution for the search cost of a random element when we construct a linear probing hash table using the Robin Hood heuristic, in tables with buckets of size b. As far as we know this is the first distributional analysis of a hashing scheme with buckets of size b. An open problem presented in the first edition of [29] requested for the average search cost of a random element in a linear probing hash table with buckets of size b. This problem was solved in [49], and this paper generalize this result.

More specifically, we give the distribution for the search cost of a random element for $b\alpha$-full tables ($0\leq\alpha<1$), as $n,m\to\infty$ while $n/m=b\alpha$. These results can also be derived for tables of fixed length m and size n, by the use of the Poisson transform presented in Section 3. However, the formulae lead to very lengthy and complicated expressions, and so we generally leave the results only in the Poisson model. It is mandatory to acknowledge that several of the main technical results needed to present this analysis have been presented in [4]. These contributions are a key component to find the generating functions for the Tuba Numbers, that lead to exact expressions for the distributions presented in this paper. More specifically in Lemma 3.1 (page 595) they characterize the sequence $T_{k,0,b}$ ($d=0$), and in Theorem 4.1 (page 602) they give its generating function. Other important related results presented in that paper are used as a starting basis for our analysis. Nevertheless, they do not exploit the combinatorial structure of

[(i)] This name was suggested by J. Ian Munro, while the author was working on his PhD. thesis the problem as presented in [15]. This combinatorial structure allows us to find powerful results based on the methodology presented in [16]. As a consequence, when we interpret the results in [4] under a combinatorial point of view, we may generalize those theorems and find the generating functions for the Tuba Numbers for all values of d.

Another main contribution of the paper (as a consequence of the methodology used to analyze the Robin Hood algorithm) is the full distribution of the number of cars that overflow in the parking problem with buckets. This problem was introduced in [34] and in [29] the problem is presented as *"A certain one-way street has m parking spaces in a row numbered 1 to m. A man and his dozing wife drive by, and suddenly, she wakes up and orders him to park immediately. He dutifully parks at the first available space [. . .]."* This problem has been extensively studied, and some later references are [39, 11, 10, 40]. A general framework for this problem can be found in [16, 29]. We have to notice that some the results presented in [39] could be derived from the results in [21, 48] by depoissonization, since the analysis of Robin Hood needs as a subproblem the analysis of the parking problem. In this paper we give the distribution for the number of cars that overflow, when $\alpha < 1$. This is the first result for the parking problem with buckets of size $b > 1$.

This paper is organized as follows. Section 2 and 3 are devoted to present some basic mathematical background used in the analysis, like the Tree Function and the Poisson Transform. It continues in Section 4 with the presentation of the Robin Hood Linear Probing Hashing Algorithm and in Section 5 with its relation with the Parking Problem. The main methodological contributions of the paper are presented in Section 6 where a combinatorial characterization of Linear Probing Hashing introduced already in [33, 15] is used to proper interpret and generalize the results presented in [4], and in Section 7 where a new sequence of numbers $T_{k,d,b}$ (called *Tuba Numbers*), which is a key tool to perform this analysis, is studied. Finally in Section 8 a distributional analysis of the Parking Problem is presented and in 4 the distributional analysis of the search cost of a random element in the Robin Hood Linear Probing Hashing is derived, including the exact expressions in the Poisson Model for all the factorial moments.

2 The tree function and the Q functions

One of the main characters in this paper is the tree function that is defined implicitly by $T(z) = ze^{T(z)}$ (and so $T(ze^{-z}) = z$) and that appears originally in problems related with the counting of rooted labeled trees [16, 22, 37, 50]. The Lagrange inversion theorem provides a number of series expansions like

$$T(z) = \sum_{n\geq 1} \frac{n^{n-1}}{n!} z^n, \qquad T(z)^m = m \sum_{n\geq m} \frac{n^{n-m-1}}{n!} n^{\underline{m}} z^n, \tag{2}$$

where $a^{\underline{k}} = a(a-1)\cdots(a-k+1)$. Most generating functions in this paper involve rational fractions in $T(z)$ with denominators that are powers of $(1-T)^{-1}$. Lagrange inversion also provides

$$\frac{1}{1-T(z)} = 1 + \sum_{n=1}^{\infty} n^n \frac{z^n}{n!}. \tag{3}$$

The asymptotic form of coefficients of any rational function of T is also directly recovered by singularity analysis [18, 38]. An application of the method requires the singular expansion of $T(z)$, itself obtained from the implicit function theorem.

Lemma 1. *The function $T(z)$ has a unique dominant singularity at $z = 1/e$, and its singular expansion in a slit neighbourhood of $1/e$ is*

$$T(z) = 1 - \delta(z) + \frac{1}{3}\delta(z)^2 - \frac{11}{72}\delta(z)^3 + \frac{43}{540}\delta(z)^4 + O\left(\delta(z)^5\right). \tag{4}$$

where $\delta(z) = \sqrt{2}\sqrt{1-ez}$.

In close association with the tree function is what Knuth has popularized under the name of the "Ramanujan Q-function". This function [3, 28, 30, 29, 16] and its close relatives play a central rôle in the analysis of many algorithms and data structures —hashing with linear probing [26, 29], union-find algorithms [32], interleaved memory [31], optimal caching [27], and random mappings [6, 17, 30], most notably. The Q-function is defined by

$$Q(n) = 1 + \frac{n-1}{n} + \frac{(n-1)(n-2)}{n^2} + \cdots,$$

or, in a way that is equivalent thanks to (2),

$$\log\frac{1}{1-T(z)} = \sum_{n\geq 0} Q(n) n^{n-1}\frac{z^n}{n!}. \tag{5}$$

Singularity analysis of the generating function yields immediately

$$Q(n) \sim \sqrt{\frac{\pi n}{2}} - \frac{1}{3} + \frac{1}{12}\sqrt{\frac{\pi}{2n}} - \frac{4}{135n} + \cdots. \tag{6}$$

An asymptotic series for $Q(n)$ was first derived by Ramanujan [3], and tight estimates are obtained in [14].

For the purpose of expressing the average-case analysis of sparse tables, Knuth [29] has extended the Ramanujan Q-function as

$$Q_r(m,n) = \sum_{i\geq 0}\binom{i+r}{i}\frac{n^{\underline{i}}}{m^i},$$

so that $Q(n) = Q_0(n, n-1)$. From the definition, one has

$$\sum_{n=0}^{\infty} Q_r(m,n) m^n \frac{t^n}{n!} = \frac{e^{mt}}{(1-t)^{r+1}}. \tag{7}$$

Then, by differentation, $Q_r(m,n) = V_r(m,n)Q_0(m,n)$, for some polynomials $V_r(m,n)$ that can be mechanically obtained from (7). For α fixed with $0 \leq \alpha < 1$, basic asymptotic approximations entail

$$\begin{aligned} Q_0(m, \alpha m - 1) \quad = \quad & \frac{1}{1-\alpha} - \frac{1}{(1-\alpha)^3}m^{-1} + \frac{2+\alpha}{(1-\alpha)^5}m^{-2} - \frac{\alpha^2+8\alpha+6}{(1-\alpha)^7}m^{-3} \\ & + \frac{\alpha^3+22\alpha^2+58\alpha+24}{(1-\alpha)^9}m^{-4} + O(m^{-5}), \end{aligned} \tag{8}$$

and

$$Q_r(m,m-c) = \left(\frac{\sqrt{\pi}}{\Gamma\left(\frac{r+2}{2}\right)2^{\frac{r+1}{2}}}\right) m^{\frac{r+1}{2}} + \left(\frac{\sqrt{\pi}}{\Gamma\left(\frac{r+1}{2}\right)2^{\frac{r}{2}}}\right)\left(\frac{r+2-3c}{3}\right) m^{\frac{r}{2}}$$
$$+ \left(\frac{\sqrt{\pi}}{\Gamma\left(\frac{r+2}{2}\right)2^{\frac{r+1}{2}}}\right)\left(\frac{3(r+2)^2-6(r+2)+2+12c^2}{24} - \frac{c(r+1)}{2}\right.$$
$$\left. + \frac{4r(r+2)^2-7r(r+2)}{72} - \frac{cr(r+1)}{3} + \frac{rc^2}{2}\right) m^{\frac{r-1}{2}} + O\left(m^{\frac{r-2}{2}}\right), \quad (9)$$

where c and r are fixed. See [42] for a general framework.

3 The Poisson Transform

There are two standard models that are extensively used in the analysis of hashing algorithms: the *exact filling* model and the *Poisson filling* model. Under the exact model, we have a fixed number of keys, n, that are distributed among m locations, and all m^n possible arrangements are equally likely to occur.

Under the Poisson model, we assume that each location receives a number of keys that is Poisson distributed with parameter λ, and is *independent* of the number of keys going elsewhere. This implies that the total number of keys, N, is itself a Poisson distributed random variable with parameter λm:

$$Pr\,[N=n] = \frac{e^{-\lambda m}(\lambda m)^n}{n!} \quad n \;=\; 0,1,\ldots$$

This model was first considered in the analysis of hashing by Fagin *et al* [12] in 1979.

Consider a hash table of size m with n elements, in which conflicts are resolved by open addressing using some heuristic. Let P be a property (e.g. cost of a successful search) of a random element of the table, and $f_{m,n}$ be the result of applying a linear operator f (e.g. an expected value) to the probability generating function of P that was found using the exact filling model. Then $\tilde{f}_m(x)$, the result of computing the same linear operator f to the probability generating function of P computed using a model with m random independent Poisson distributed objects each with parameter x, is

$$\tilde{f}_m(x) = \sum_{n\geq 0} Pr\,[N=n]\, f_{m,n} = e^{-mx}\sum_{n\geq 0}\frac{(mx)^n}{n!} f_{m,n}. \quad (10)$$

We may use (10) to define $\mathbf{P}_m[f_{m,n};x]$, the *Poisson transform* (also called *Poisson generating function* [13, 24]) of $f_{m,n}$, as

$$\mathbf{P}_m[f_{m,n};x] = \tilde{f}_m(x) = e^{-mx}\sum_{n\geq 0}\frac{(mx)^n}{n!} f_{m,n}. \quad (11)$$

If $\mathbf{P}_m[f_{m,n};\lambda]$ has a MacLaurin expansion in powers of λ, then we can retrieve the original sequence $f_{m,n}$ by the following inversion theorem [21]:

Theorem 1 (Depoissonization Theorem)**.** *If* $\mathbf{P}_m[f_{m,n};\lambda] = \sum_{k\geq 0} a_{m,k}\lambda^k$ *then* $f_{m,n} = \sum_{k\geq 0} a_{m,k}\frac{n^{\underline{k}}}{m^k}$.

In this paper we consider $\lambda = b\alpha$.

The results obtained under the Poisson filling model can also be interpreted as an approximation of those one would obtain under the exact filling model when $n, m \to \infty$ with $n = b\alpha m$. This approximation can be formalized by means of an asymptotic expansion. Poblete, in [42], presents an approximation theorem and gives an explicit form for all the terms of the expansion.

Theorem 2 ([42])**.** *For $b\alpha = n/m$,*

$$f_{m,n} = \mathbf{P}_m[f_{m,n}; b\alpha] + \sum_{j\geq 1} \left(\frac{1}{n}\right)^j \sum_{i=j+1}^{2j} (b\alpha)^i c_{i,j} \partial_\alpha^i \mathbf{P}_m[f_{m,n}; b\alpha],$$

where

$$c_{i,j} = \frac{1}{i!} \sum_{k\geq 0} (-1)^{i-k+j} \binom{i}{k} \begin{bmatrix} k \\ k-j \end{bmatrix},$$

and $\left[{k \atop k-j}\right]$ denotes the Stirling numbers of the first kind.

For most situations and applications, this approximation is satisfactory. However, it cannot be used when we have a full, or almost full table (α is very close to 1).

4 The Robin Hood Linear Probing Hashing Algorithm

We follow the ideas presented in [48] and [49]. Figure 1 shows the result of inserting elements with the keys 36, 77, 24, 69, 18, 56, 97, 78, 49, 79, 38 and 10 in a table with ten buckets of size 2, with hash function $h(x) = x$ mod 10, and resolving collisions by linear probing using the Robin Hood heuristic.

	0	1	2	3	4	5	6	7	8	9
a	69	10			24		36	77	18	78
	79						56	97	38	49

Fig. 1: A Robin Hood Linear Probing hash table.

When there is a collision in location i, then the element that has probed the least number of locations, probes location $(i + 1)$ mod m. In the case of a tie, we (arbitrarily) move the element whose key has largest value.

Figure 2 shows the partially filled table after inserting 58. There is a collision with 18 and 38. Since there is a tie (all of them are in their first probe location), we arbitrarily decide to move 58, the largest key. Then 58 is in its second probe location, 78 also, but 49 is in its first one. So 49 has to move. Then 49, 69, 79 are all in their second probe location, so 79 has to move to its final position by the tie-break policy described above.

The following properties are easily verified:

- At least one element is in its home location.

a	49	79			24		36	77	18	58
	69	10					56	97	38	78
	0	1	2	3	4	5	6	7	8	9

Fig. 2: The table after inserting 58.

- The keys are stored in nondecreasing order by hash value, starting at some location k and wrapping around. In our example k=4 (corresponding to the home location of 24).
- If a fixed rule (that depends only on the value of the keys and not in the order they are inserted) is used to break ties among the candidates to probe their next probe location (eg: by sorting these keys in increasing order), then the resulting table is independent of the order in which the elements were inserted [8].

5 Linear Probing Sort and the Parking Problem

To analyze Robin Hood linear probing, we first have to discuss some ideas presented in [7, 48, 49] and [21]. When the hash function is order preserving (that is, if $x < y$ then $h(x) < h(y)$), a variation of the Robin Hood linear probing algorithm can be used to sort [21], by successively inserting the n elements in an initially empty table. In this case, instead of letting the excess elements from the rightmost location of the table wrap around to location zero, we can use an *overflow area* consisting of locations m, $m+1$, etc. The number of locations needed for this overflow area is an important performance measure for this sorting algorithm.

This problem is related to the study of the cost of successful searches in the Robin Hood linear probing algorithm, as follows. Without loss of generality, we search for an element that hashes to location 0. Moreover, since the order of the insertion is not important, we assume that this element is the last one inserted. If we look at the table after the first n elements have been inserted, all the elements that hash to location 0 (if any) will be occupying contiguous locations, near the beginning of the table. The locations preceding them will be occupied by elements that wrapped around from the right end of the table, as can be seen in Figure 2. The key observation here is that those elements are exactly the ones that would have gone to the overflow area. Furthermore, it is easy to see that the number of elements in this overflow area does not change when the elements that hash to 0 are removed. As a consequence, the cost of retrieving an elements that hashes to 0 can be divided in two parts.

- The number of elements that wrap around the table. In other words, the size of the overflow area.
- The number of elements that hash to location 0.

In Section 8 we study the distribution of the number of the elements that overflow, and in Section 9 we study the distribution for the successful search cost of a random element. In this setting, linear probing sort is the parking problem introduced in [34].

In this paper we give the distribution for the number of cars that overflow when $\alpha < 1$, generalizing to $b > 1$ some of the results presented in [48, 21, 39] for the case $b = 1$. We thank very much to an anonymous referee for pointng us out that in [46] an exact formula for the generating function of the overflow (called defective bucket parking functions) in the exact filling model has been derived by a rather simple and direct approach. Thus in that thesis an equivalent of the main theorem of Section 8 (Theorem 9) is presented.

We work in the Poisson model, since the presentation is much simpler than in the exact model. Then, with the use of the Poisson transform, we may obtain the exact results for fixed m, n, and b. It is important to note that in [21, 39, 48] the exact distribution of the elements that overflow when $b = 1$ and in [49] the average number of elements that overflow (for general b) are calculated in the exact model. This analysis brings a new point of view on the problem. The key element to solve this problem is the use of a new sequence of numbers that we call *Tuba Numbers*

6 Combinatorial characterization of Linear Probing Hashing

Under a combinatorial point of view Linear Probing can be seen as a *sequence* of *almost full tables* [15].

10	70		33	24		36		18	78
30	81		63			56		38	49
40			73			96		58	
0	1	2	3	4	5	6	7	8	9

Fig. 3: A sequence of six almost full tables starting at locations 0, 2, 3, 5, 6, and 8.

By circular symmetry [29] and for nonfull tables ($n < bm$), we may freely assume that one of the empty locations belongs to the rightmost bucket. This assumption of a last empty location in nonfull tables is made from now onwards. When all the empty places belong to the last bucket of the table we say that such a table is almost full. For a given length m there are b almost full tables since the last bucket may contain from 0 to $b - 1$ empty locations.

As a consequence, a general table decomposes as a labeled product of clusters (sometimes also figuratively called "islands") that are, up to relabeling, almost full tables. Furthermore, it will be enough to study almost full tables, and then generalize the results for general tables in a similar way as it is done in [15], by using the sequence construction for labelled structures as presented in [16].

As an example, the hash table in Figure 3 has buckets of size $b = 3$ and six clusters (almost full tables) starting at locations 0, 2, 3, 5, 6, and 8. In a respectively order, their lengths are 2, 1, 2, 1, 2, and 2; their sizes are 5, 0, 4, 0, 3, and 5; and they have 1, 3, 2, 3, 3, 1 free locations in their last bucket.

We present now a combinatorial interpretation of several results presented in [4], that will bring light to some important generalizations. Let F_{bn+d} be the number of ways to construct an almost full table of length $n + 1$ and size $bn + d$ (that is, there are $b - d$ empty slots in the last bucket). Define also

$$F_d(u) = \sum_{n \geq 0} F_{bn+d} \frac{u^{bn+d}}{(bn+d)!} \qquad N_d(z, w) = \sum_{s=0}^{b-1-d} w^{b-s} F_s(zw), \quad 0 \leq d \leq b - 1. \tag{12}$$

In this setting $N_d(z,w)$ is the generating function for the number of almost full tables with more than d empty locations in the last bucket.

The *elementary symmetric functions* of variables $\gamma_j(z)$ are defined as the coefficients $\{\sigma_k(z)\}$ of the polynomial $\sum_{k=0}^{b}\sigma_k(z)x^{n-k} = \prod_{j=0}^{b-1}(x+\gamma_j(z))$. Let r be a primitive b-th root of unity and $\sigma_k(z)$ be the k-th elementary symmetric function of the variables $\{T(r^j z), 0 \le j < b\}$, where T is the Tree function. Lemma 2.3 (page 594) in [4] states

$$(bz)^{b-d}F_d(bz) = (-1)^{b-d-1}b^{b-d}\sigma_{b-d}(z), \tag{13}$$

and formula 3.8 (page 597) states

$$N_0(z,w) = 1 - \prod_{i=0}^{b-1}\left(1 - \frac{b}{z}T\left(r^i\frac{zw}{b}\right)\right). \tag{14}$$

Moreover, since $T(\alpha e^{-\alpha}) = \alpha$, then

$$N_0(b\alpha, e^{-\alpha}) = 1, \tag{15}$$

since the product is 0 because when $i = 0$, the factor is 0.

Let also $Q_{m,n,d}$ be the number of ways of inserting n elements into a table with m buckets of size b, so that a given (say the last) bucket of the table contains more than d empty slots. The bucket size b remains fixed, so we do not include it as a subscript in the sequel. There cannot be more empty slots than the size of the bucket so $Q_{m,n,b} = 0$. For each of the m^n possible arrangements, the last bucket has 0 or more empty slots, and so $Q_{m,n,-1} = m^n$. Observe that $Q_{m,n,0}$ gives the number of ways of inserting n elements into a table with m buckets, so that the last bucket is not full. For notational convenience, we define $Q_{0,n,d} = [n = 0]$ (following the notation presented in [23] we use $[S]$ to represent 1 if S is true, and 0 otherwise). Let also define

$$\Lambda_d(z,w) = \sum_{m\ge 0}\sum_{n\ge 0} Q_{m,n,d}\frac{z^n}{n!}w^{bm}.$$

Then, $\Lambda(z,w)$ is the generating function for the number of ways to construct hash tables such that their last bucket is not full. After a somehow tedious calculation, Lemma 3.2 (page 597) in [4] states

$$\Lambda_0(z,w) = \frac{N_0(z,w)}{1-N_0(z,w)}. \tag{16}$$

Identity (16) could be directly derived, by interpreting a general hash table with more than 0 empty locations in its last bucket as a non-empty sequence of almost full tables, all of them with more than 0 empty locations in their last bucket. The generating function for $\Lambda_0(z,w)$ then follows by standard combinatorial techniques as presented in [16].

This combinatorial interpretation allows a natural generalization of these results, to find $\Lambda_d(z,w)$ for all $0 \le d \le b-1$: a general hash table with more than d empty slots in its last bucket is a sequence of almost full table with more than 0 empty locations, followed by an almost full table with more than d empty slots. As a consequence, we directly deduce:

Lemma 2.

$$\Lambda_d(z,w) = \frac{N_d(z,w)}{1-N_0(z,w)}.$$

We may find explicit expressions for these generating functions with the use of the Tuba Numbers.

7 Tuba Numbers

It does not seem possible to find a closed formula for $Q_{m,n,d}$. This sequence of numbers is important since $(Q_{m,n,b-d+1} - Q_{m,n,b-d})/m^n$ is the probability that a given bucket of the table contains exactly d elements. In [36], Mendelson proves

Theorem 3. *For* $0 \le d \le b-1$, *and* $m > 0$,

$$Q_{m,n,d} = \begin{cases} \sum_{j=0}^{n} \binom{n}{j} Q_{m-1,j,d} & 0 \le n < mb\text{-}d \\ 0 & n \ge mb\text{-}d \end{cases}$$

However, a new approach to the study of the numbers $Q_{m,n,d}$, is presented in [49], where a new sequence of numbers $T_{k,d,b}$ is introduced that satisfies a recurrence resembling that of the Bernoulli numbers. This new sequence may be helpful in solving problems involving recurrences with truncated generating functions.

Let

$$[A(z)]_n = \sum_{i=0}^{n} a_i u^i,$$

then Theorem 3 translates into the recurrence relation

$$\begin{aligned} Q_{0,d}(u) &= 1, \\ Q_{m,d}(u) &= [e^u Q_{m-1,d}(u)]_{bm-d-1}, \quad m \ge 1, \end{aligned} \tag{17}$$

for $Q_{m,d}(u) = \sum_{n \ge 0} Q_{m,n,d} \frac{u^n}{n!}$. The main problem is that we are dealing with a recurrence that involves truncated generating functions.

The strategy presented in [49] consists in finding an exponential generating function $T_d(u)$ such that

$$Q_{m,d}(u) = [T_d(u)e^{mu}]_{bm-d-1} \tag{18}$$

where $T_d(u) = \sum_{k \ge 0} T_{k,b,d} \frac{u^k}{k!}$, for some coefficients $T_{k,b,d}$ to be determined, and independent of m. Here, b is an implicit parameter, and we use the expression $T_{k,d}$.

The intuition behind this idea is as follows. From (17), we obtain $Q_{m,d}(u)$ by multiplying the truncated generating function $Q_{m-1,d}(u)$ by the series e^u and then taking only the first $bm - d$ terms of it (monomials of degree 0, 1, up to , $bm - d - 1$). Moreover, $Q_{0,d}(u)$ is the first term of e^u. It is clear that without any truncations $Q_{m,d}(u)$ would be e^{mu}. However we have to consider a correcting factor originated by these truncations and this is the reason for defining this generating function $T_d(u)$. Then (18) gives a non recursive definition of $Q_{m,d}(u)$ that involves the truncated product of two series. The interesting aspect of this approach is that $T_d(u)$ does not depend on m. Furthermore, the only dependency on m is captured in the well known series that converges to e^{mu}.

The Tuba Numbers $T_{k,d}$ satisfy some nice properties. For example, the next two theorems are proved in [47]. The following can indeed be used as definition.

Theorem 4.

$$\sum_j \binom{k}{j} \left(\left\lfloor \frac{k+d}{b} \right\rfloor \right)^{k-j} T_{j,d} = [k=0]. \tag{19}$$

A very curious property of these numbers is

Theorem 5.

$$\sum_{d=0}^{b-1} T_{k,d} = \begin{cases} b & k = 0, \\ -1 & k = 1, \\ 0 & k > 1, \end{cases} \tag{20}$$

that translates into

$$\sum_{d=0}^{b-1} T_d(b\alpha) = \sum_{d=0}^{b-1} \left(\sum_{k \geq 0} T_{k,d} \frac{(b\alpha)^k}{k!} \right) = b(1-\alpha). \tag{21}$$

Equation (21) is very useful to simplify several expressions in the analysis. In this paper we generalize equation (21). One of the key observations is that $T_d(b\alpha)$ is the limit of the Poisson transform of $Q_{m,n,d}/m^n$ (the probability that a given bucket contains more than d empty slots) when $m, n \to \infty$, $n = b\alpha m$ and $\alpha < 1$ since

$$\begin{aligned} \lim_{m\to\infty} \mathbf{P}_m[Q_{m,n,d}/m^n; b\alpha] &= \lim_{m\to\infty} e^{-mb\alpha} \sum_{n\geq 0} \frac{Q_{m,n,d}}{m^n} \frac{(mb\alpha)^n}{n!} \\ &= \lim_{m\to\infty} e^{-mb\alpha} \left[T_d(b\alpha) e^{mb\alpha} \right]_{bm-d-1} = T_d(b\alpha). \end{aligned}$$

As a consequence $T_d(b\alpha)$ is the probability that a random bucket has more than d empty locations when $m, n \to \infty$, $n = b\alpha m$ and $\alpha < 1$. The rate of convergence to this limit value is exponentially small.

Theorem 6. *Let $\Upsilon_{m,d}(b\alpha) = \mathbf{P}_m[Q_{m,n,d}/m^n; b\alpha]$. That is, $\Upsilon_{m,d}(b\alpha)$ is the probability, in the Poisson Model, that a given bucket contains more than d empty slots when $bm\alpha$ elements are inserted in a hash table with m buckets of size b, using linear probing as collision resolution scheme. Then, when $\alpha < 1$,*

$$\Upsilon_{m,d}(b\alpha) = T_d(b\alpha) + O\left(\alpha^{-d} e^{bm(1-\alpha+\log\alpha)} m^{-3/2} \right).$$

Proof: We first have to notice that when $\alpha < 1$, then $1 - \alpha + \log\alpha < 0$. Since $\Upsilon_{m,d}(b\alpha) = e^{-mb\alpha} \left[T_d(b\alpha) e^{mb\alpha} \right]_{bm-d-1}$ and $T_d(b\alpha) \leq \sum_{d=0}^{b-1} T_d(b\alpha) = b(1-\alpha)$ then

$$\begin{aligned} | T_d(b\alpha) - \Upsilon_{m,d}(b\alpha) | &\leq | e^{-mb\alpha} b(1-\alpha) \sum_{j=bm-d}^{\infty} \frac{(mb\alpha)^n}{n!} | \\ &= e^{-mb\alpha} \frac{(mb\alpha)^{mb-d}}{(mb-d)!} \sum_{j=0}^{\infty} \left(\frac{j-d}{m} \right) \frac{(mb\alpha)^j}{(j+mb-d)^{\underline{j}}} \\ &\leq e^{-mb\alpha} \frac{(mb\alpha)^{mb-d}}{(mb-d)!} \sum_{j=0}^{\infty} \left(\frac{j-d}{m} \right) \left(\frac{mb\alpha}{mb-d+1} \right)^j . \end{aligned}$$

Since the last sum converges when $\alpha < 1$, then

$$| T_d(b\alpha) - \Upsilon_{m,d}(b\alpha) |= O\left(e^{-mb\alpha} \frac{(mb\alpha)^{mb-d}}{(mb-d)!} \right) = O\left(\alpha^{-d} e^{bm(1-\alpha+\log\alpha)} m^{-3/2} \right). \qquad \square$$

Theorem 6 generalizes Theorem 3.1 in [21] since the latter only considers the case $b = 1$ and $d = 0$. The identity $\Upsilon_{m,d}(b\alpha) = e^{-mb\alpha}\left[T_d(b\alpha)e^{mb\alpha}\right]_{bm-d-1}$ leads to the following lemma that will be used in the proof of Theorem 9.

Lemma 3.
$$\Upsilon_{m,d}(b\alpha) = T_d(b\alpha) + O\left((b\alpha)^{bm-d}\right).$$

The rest of this section is devoted to find an explicit expression for $T_d(b\alpha)$. The generating function $T_0(b\alpha)$ has already been studied in [4]:

Theorem 7 (Theorem 4.1 in [4])**.**

$$T_0(b\alpha) = \frac{b(1-\alpha)}{\prod_{j=1}^{b-1}\left(1 - \frac{T(r^j\alpha e^{-\alpha}}{\alpha}\right)},$$

where T is the Tree function and r is a b-th root of unity.

This result can be generalized for $T_d(b\alpha)$ for all $0 \le d \le b-1$. We need first to prove an important Lemma.

Lemma 4.
$$T_d(b\alpha) = N_d(b\alpha, e^{-\alpha})T_0(b\alpha), \quad 0 \le d \le b-1.$$

Proof: The decomposition of a general full table as a sequence of almost full tables leads to

$$Q_{m,n,d} = \sum_{j\ge 0}\sum_{s=0}^{b-d-1}\binom{n}{bj+s}F_{bj+s}Q_{m-1-j,n-j-bs,0}. \tag{22}$$

The n elements inserted can be divided in the $bj+s$ elements that go into the last cluster of length $j+1$ and $n-bj-s$ elements that go in the rest of the table of length $m-1-j$. Since the last cluster should have more than d empty slots, then s should be less than $b-d$. We then have to multiply for the number of ways to construct these partial tables (F_{bj+s} and $Q_{m-1-j,n-bj-s,0}$), and sum over all possible values of j and s. If we then divide by m^n and apply the Poisson Transform to both sides of (22) we have

$$\begin{aligned}
T_d(b\alpha) &= \lim_{m\to\infty}\mathbf{P}_m[Q_{m,n,d}/m^n; b\alpha] \\
&= \lim_{m\to\infty} e^{-mb\alpha}\sum_{s=0}^{b-1-d}\sum_{j\ge 0}F_{bj+s}\frac{(b\alpha)^{bj+s}}{(bj+s)!}\sum_{n\ge bj+s}Q_{m-1-j,n-j-b-s,0}\frac{(b\alpha)^{n-bjn-s}}{(n-bj-s)!} \\
&= \lim_{m\to\infty} e^{-mb\alpha}\sum_{s=0}^{b-1-d}\sum_{j\ge 0}F_{bj+s}\frac{(b\alpha)^{bj+s}}{(bj+s)!}\sum_{n\ge 0}Q_{m-1-j,n,0}\frac{(b\alpha)^n}{n!} \\
&= \lim_{m\to\infty} e^{-mb\alpha}\sum_{s=0}^{b-1-d}\sum_{j\ge 0}F_{bj+s}\frac{(b\alpha)^{bj+s}}{(bj+s)!}\left(\left[T_0(b\alpha)e^{(m-1-j)b\alpha}\right]_{bm-d-1}\right) \\
&= T_0(b\alpha)\sum_{s=0}^{b-1-d}\left(e^{-\alpha}\right)^{b-s}\sum_{j\ge 0}F_{bj+s}\frac{(b\alpha)^{bj+s}}{(bj+s)!}\left(e^{-\alpha}\right)^{bj+s} \\
&= T_0(b\alpha)N_d(b\alpha, e^{-\alpha}).
\end{aligned}$$

□

Lemma 4 relates $T_0(b\alpha)$ with $T_d(b\alpha)$. Notice that since by equation (15) $N_0(b\alpha, e^{-\alpha}) = 1$ then Lemma 4 is also valid for $d = 0$. A generalization of equation (15) to $N_d(b\alpha, e^{-b\alpha})$ will then lead to explicit expressions for $T_d(b\alpha)$. We use again elementary symmetric functions.

Lemma 5. $$N_d(b\alpha, e^{-\alpha}) = [u^d]\prod_{i=1}^{b-1}\left(1 - u\frac{T(r^i\alpha e^{-\alpha})}{\alpha}\right), \quad 0 \le d \le b-1.$$

Proof: By equation (13)

$$(bz)^{b-s}F_s(bz) = [t^s]\left(t^b - \prod_{i=0}^{b-1}\left(t - bT(r^i z)\right)\right), \quad 0 \le s \le b-1.$$

As a consequence

$$\begin{aligned}
(e^{-\alpha})^{b-s}F_s(b\alpha e^{-\alpha}) &= (b\alpha)^{s-b}[t^s]\left(u^b - \prod_{i=0}^{b-1}\left(t - bT(r^i\alpha e^{-\alpha})\right)\right)\\
&= -(b\alpha)^s[t^s]\prod_{i=0}^{b-1}\left(\frac{t}{b\alpha} - \frac{T(r^i\alpha e^{-\alpha})}{\alpha}\right)\\
&= -[u^s]\prod_{i=0}^{b-1}\left(u - \frac{T(r^i\alpha e^{-\alpha})}{\alpha}\right).
\end{aligned}$$

We then have

$$\begin{aligned}
N_d(b\alpha, e^{-\alpha}) &= \sum_{s=0}^{b-1-d}(e^{-\alpha})^{b-s}F_s(b\alpha e^{-\alpha}) = -\sum_{s=0}^{b-1-d}[u^s]\prod_{i=0}^{b-1}\left(u - \frac{T(r^i\alpha e^{-\alpha})}{\alpha}\right)\\
&= -[u^{b-1-d}]\frac{\prod_{i=0}^{b-1}\left(u - \frac{T(r^i\alpha e^{-\alpha})}{\alpha}\right)}{1-u} = -[u^{b-1-d}]\frac{(u-1)\prod_{i=1}^{b-1}\left(u - \frac{T(r^i\alpha e^{-\alpha})}{\alpha}\right)}{1-u}\\
&= [u^{b-1-d}]\prod_{i=1}^{b-1}\left(u - \frac{T(r^i\alpha e^{-\alpha})}{\alpha}\right) = [u^d]\prod_{i=1}^{b-1}\left(1 - u\frac{T(r^i\alpha e^{-\alpha})}{\alpha}\right).
\end{aligned}$$

□

Lemmas 4 and 5 lead to the main theorem of this section:

Theorem 8.

$$T_d(b\alpha) = b(1-\alpha)\frac{[u^d]\prod_{i=1}^{b-1}\left(1 - u\frac{T(r^i\alpha e^{-\alpha})}{\alpha}\right)}{\prod_{j=1}^{b-1}\left(1 - \frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}, \quad 0 \le d \le b-1,$$

where T is the Tree function and r is a b-th root of unity.

The following Corollary will be very useful in the analysis of the parking problem with buckets.

Corollary 1.

$$\sum_{d=0}^{b-1}T_{b-1-d}(b\alpha)z^d = b(1-\alpha)\frac{\prod_{j=1}^{b-1}\left(z - \frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}{\prod_{j=1}^{b-1}\left(1 - \frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}.$$

8 The parking problem with buckets

Notation. Given a bivariate function $G(z,\alpha)$ we define

$$\mathrm{Z}G(z,\alpha)=G(0,\alpha)\qquad \mathrm{U_z}G(z,\alpha)=G(1,\alpha)\qquad \partial_z G(z,\alpha)=\frac{\partial G(z,\alpha)}{\partial z}.$$

We first state the main Theorem of this section.

Theorem 9. *Let $w_{m,b\alpha,k}$ be the probability of having k cars going to overflow in a $b\alpha$-full table with m buckets of size b and $\alpha<1$, and $\Omega_m(b\alpha,z)=\sum_{k\geq 0} w_{m,b\alpha,k}z^k$. Then*

$$\Omega_m(b\alpha,z)=\left(\frac{b(1-\alpha)(z-1)}{z^b-e^{b\alpha(z-1)}}\right)\frac{\prod_{j=1}^{b-1}\left(z-\frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}{\prod_{j=1}^{b-1}\left(1-\frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}+O\left(\alpha^{bm}\right). \tag{23}$$

Remark 1. When $b=1$ then, in the terminology of [39], $w_{mb\alpha,k}$ is the probability of having a *defective parking function of defect k.*

Proof of Theorem 9: The proof follows closely the ideas presented in [21]. We first derive a recurrence for $\Omega_m(b\alpha,z)$, and then solve it. Let define

$$\Omega(b\alpha,z)=\left(\frac{b(1-\alpha)(z-1)}{z^b-e^{b\alpha(z-1)}}\right)\frac{\prod_{j=1}^{b-1}\left(z-\frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}{\prod_{j=1}^{b-1}\left(1-\frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}.$$

Since we have a Poisson process, with probability $e^{-b\alpha}\frac{\alpha^k}{k!}$ the last bucket receives, in addition to the elements that overflow from the previous bucket, k elements that hash to it. From these elements, all but b of them go to overflow, and their contribution to the recurrence is

$$e^{b\alpha(z-1)}\frac{\Omega_{m-1}(b\alpha,z)}{z^b}. \tag{24}$$

However, when the last bucket receives less than b elements in total, there is no overflow, and so we need a correction term. This correction term is

$$\sum_{s=1}^{b}\left(1-z^{-s}\right)P_{m,s}(b\alpha),$$

where $P_{m,s}(b\alpha)$ is the probability (in the Poisson Model) of having $b-s$ elements in the last bucket. As we have seen in Section 7 this value is equal to $\Upsilon_{m,s-1}(b\alpha)-\Upsilon_{m,s}(b\alpha)$, and so the contribution of this correction term is

$$\sum_{s=1}^{b}\left(1-z^{-s}\right)\left(\Upsilon_{m,s-1}(b\alpha)-\Upsilon_{m,s}(b\alpha)\right)=\left(1-z^{-1}\right)\sum_{s=0}^{b-1}\Upsilon_{m,s}(b\alpha)z^{-s}. \tag{25}$$

Notice that $\Upsilon_{m,b}(b\alpha) = 0$ since $Q_{m,n,b} = 0$ because there cannot be more than b empty locations in a bucket. As a consequence, from equations (24) and (25) we obtain the following recurrence

$$\Omega_m(b\alpha, z) = e^{b\alpha(z-1)} \frac{\Omega_{m-1}(b\alpha, z)}{z^b} + \left(1 - z^{-1}\right) \sum_{s=0}^{b-1} \Upsilon_{m,s}(b\alpha) z^{-s}. \tag{26}$$

If we can establish that the sequence $\Omega_m(b\alpha, z)$ converges to $\Omega(b\alpha, z)$ when $m \to \infty$, then we finally get

$$\Omega(b\alpha, z) = \frac{(z-1)}{z^b - e^{b\alpha(z-1)}} \sum_{s=0}^{b-1} T_{b-1-s}(b\alpha) z^s. \tag{27}$$

The result would then follow by equation (27) and Corollary 1.

The rest of the proof is devoted to prove the existence of such a limit. Let $M(b\alpha, z) = \frac{e^{b\alpha(z-1)}}{z^b}$, $N(z) = 1 - z^{-1}$, and $R_m(b\alpha, z) = \sum_{s=0}^{b-1} \Upsilon_{m,s}(b\alpha) z^{-s}$. Then, by equation (26) we have

$$\Omega_m(b\alpha, z) = M(b\alpha, z)^m + N \sum_{i=1}^{m} M(b\alpha, z)^{m-i} R_i(b\alpha, z). \tag{28}$$

As a consequence, when $\alpha < 1$, the convergence from $\Omega_m(b\alpha, z)$ to $\Omega(b\alpha, z)$ is established by equation (28) and Theorem 6. It remains to prove that for $n = 0 \ldots bm - 1$ ($\alpha < 1$), then $[\alpha^n]\Omega_m(b\alpha, z) = [\alpha^n]\Omega(b\alpha, z)$.

By equation (27) and Lemma 3 we have

$$\Omega_m(b\alpha, z) = M(b\alpha, z)^m + N \sum_{s=0}^{b-1} \left(T_s(b\alpha) + O\left(\alpha^{bi-s}\right)\right) z^{-s} \sum_{i=1}^{m} M(b\alpha, z)^{m-i}.$$

Moreover,

$$M(b\alpha, z)^{m-i} = \sum_{n \geq 0} \frac{((m-i)b\alpha)^n}{n!} \sum_{k \geq 0} \binom{n}{k} (-1)^{n-k} z^{k-b(m-i)},$$

and so $[\alpha^n]M(b\alpha, z)^{m-i}$ contributes with powers of z from $-b(m-i)$ to $n - b(m-i)$ to $[\alpha^n]\Omega_m(b\alpha, z)$.

Since $[\alpha^j]\ \Upsilon_{m,s}(b\alpha) = [\alpha^j]\ T_s(b\alpha)$ for $j = 0 \ldots bm - s - 1$, then

$$\begin{aligned} [z^p][\alpha^n]\ (\Upsilon_{m,s}(b\alpha) M(b\alpha, z)^{m-i} z^{-s}) &= [z^p][\alpha^n]\ (T_s(b\alpha) M(b\alpha, z)^{m-i} z^{-s}), \\ & p = n - bm + 1 \ldots n - bm + bi - s. \end{aligned}$$

As a consequence, by considering all the contributions in the intersection of all these ranges of p when $i = 1..m$, we have

$$\begin{aligned} [z^p][\alpha^n]\ N \sum_{s=0}^{b-1} T_s(b\alpha) z^{-s} \sum_{i=1}^{m} M(b\alpha, z)^{m-i} &= \\ [z^p][\alpha^n]\ N \sum_{s=0}^{b-1} \Upsilon_{m,s}(b\alpha) z^{-s} \sum_{i=1}^{m} M(b\alpha, z)^{m-i}, &\quad p = n - bm + 1 \ldots n - b + 1. \end{aligned}$$

On the other hand, it is not difficult to see that

$$[z^p][\alpha^n]\,(\Omega(b\alpha, z) - \Omega_m(b\alpha, z)) = 0, \quad p = n - bm + 1 \dots n - b + 1,$$

and so

$$[z^p][\alpha^n]\,\Omega(b\alpha, z) = [z^p][\alpha^n]\,\Omega_m(b\alpha, z), \quad p = n - bm + 1 \dots n - b + 1,$$

We have then proved that when $n \leq bm - 1$ ($\alpha < 1$), then $[\alpha^n]\,\Omega(b\alpha, z) = [\alpha^n]\,\Omega_m(b\alpha, z)$. We remark than when $n \geq bm$, then this derivation only guarantees equality from powers of z that are greater to zero ($n - bm + 1 > 0$), and so $[\alpha^n]\,\Omega(b\alpha, z)$ and $[\alpha^n]\,\Omega_m(b\alpha, z)$ may differ. □

It is very important to notice that Theorem 9 allows the use of Theorem 1 with $\Omega(b\alpha, z)$ to obtain $w_{m,n,k}$ in the exact filling model when $n \leq bm - 1$, since it only needs coefficients of powers of $b\alpha$ less than n. In any case, when $n \geq bm$ the range of potential non-zero coefficients of powers of z is included in $n - bm + 1 \dots n - b + 1$, so, at least formally, the use of Theorem 1 would return the exact probabilities in the exact filling model also for $n \geq bm$.

When $b = 1$ we rederive the result presented in [48]

$$\Omega(\alpha, z) = \frac{(z-1)(1-\alpha)}{z - e^{\alpha(z-1)}}. \tag{29}$$

Moreover, some of the results presented in [39] could be derived from the results in [48, 21] by depoissonization. The probabilities can be extracted from equation (29), by expanding powers of z, and then the results in the exact filling model follow from depoissonization, after expanding those probabilities in a power series in α. An alternative approach would be to depoissonize the distribution (29) and then extract the coefficients.

For general b however, the depoissonization is rather complicated by the expressions related with the Tuba Numbers. As a consequence we present the results in the Poisson Model, and translate them in the exact model when the final expressions are relatively simple.

The first b probabilities are given in the following theorem:

Theorem 10.

$$w_{m,b\alpha,k} = \frac{e^{b\alpha}}{k!}\sum_{i=0}^{k}\binom{k}{i}(-1)^{k-i}\alpha^{k-i-1}(k-i+b\alpha)i!T_{b-1-i}(b\alpha) + O\left(\alpha^{bm}\right) \quad 0 \leq k \leq b-1.$$

More specifically, the probability that no car overflows is $w_{b\alpha,0} = e^{b\alpha}T_{b-1}(b\alpha)$.

Proof: The result follows from the known identity

$$\partial_z^k(A(z,b\alpha)B(z,b\alpha)) = \sum_{i=0}^{k}\binom{k}{i}\partial_z^{k-i}A(z,b\alpha)\,\partial_z^i B(z,b\alpha),$$

with

$$A(z,b\alpha) = \frac{(z-1)}{z^b - e^{b\alpha(z-1)}}, \qquad B(z,b\alpha) = \sum_{s=0}^{b-1}T_{b-1-s}(b\alpha)z^s,$$

and noticing that for $0 \le j \le b-1$,

$$\mathrm{Z}\partial_z^j A(z, b\alpha) = (-1)^j \alpha^{j-1}(j + b\alpha)e^{b\alpha} \quad \text{and} \quad \mathrm{Z}\partial_z^j B(z, b\alpha) = j! T_{b-1-j}(b\alpha).$$

□

When $b = 1$ then $T_0(\alpha) = (1 - \alpha)$, and then we rederive the known result $w_{\alpha,0} = e^{\alpha}(1 - \alpha)$ presented in [34]. More generally we may expand $A(z, b\alpha)$ to obtain

$$\begin{aligned} \frac{(z-1)}{z^b - e^{b\alpha(z-1)}} &= (1-z) \sum_{j \ge 0} z^{bj} e^{b\alpha(j+1)(1-z)} \\ &= \sum_{n \ge 0} \left(\sum_{j=0}^{\lfloor \frac{n}{b} \rfloor} e^{b\alpha(j+1)} \frac{(-1)^{n-bj}(b\alpha(j+1))^{n-1-bj}}{(n-bj)!} (b(\alpha - j(1-\alpha)) + n) \right) z^n. \end{aligned} \tag{30}$$

Notice, that in Theorem 10 we use expansion (30) only for $n < b$ and so $j = 0$. We may now find all the set of probabilities by using expansion (30), to obtain

Theorem 11. *For all $k \ge 0$ we have*

$$\begin{aligned} w_{m,b\alpha,k} &= \sum_{j=0}^{\lfloor \frac{k}{b} \rfloor} e^{b\alpha(j+1)} \sum_{i=0}^{\min(b-1,k)} \frac{(-1)^{k-i-bj}}{(k-i-bj)!} \\ &\qquad (b\alpha(j+1))^{k-i-1-bj}(k - i + b\alpha(j+1) - bj) T_{b-1-i}(b\alpha) + O\left(\alpha^{bm}\right). \end{aligned}$$

It does not seem possible to find a simple way to do depoissonization to the coefficients $w_{m,n,k}$. From equation (23) we may also find the expected number of cars that overflow from a random bucket.

Theorem 12. *Let $\Omega_{m,b\alpha}$ be the r.v for the number of cars that overflow from a $b\alpha$-full table with m buckets of size b and $\alpha < 1$. Then*

$$\mathbf{E}[\Omega_{m,b\alpha}] = \frac{1}{2}\left(\frac{1}{1-\alpha} - b(1+\alpha)\right) + \sum_{d=1}^{b-1} \frac{1}{\left(1 - \frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)} + O\left(\alpha^{bm}\right).$$

Proof: The result follows by taking derivatives with respect to z in equation (23), and noticing that

$$\mathrm{U_z}\partial_z \left(\frac{\prod_{j=1}^{b-1}\left(z - \frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}{\prod_{j=1}^{b-1}\left(1 - \frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)} \right) = \sum_{d=1}^{b-1} \frac{1}{\left(1 - \frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}. \tag{31}$$

□

We may expand the generating functions of the quasi-inverses of the Tree functions, and then use the depoissonization theorem. As a consequence we obtain:

Corollary 2. *Let $\Omega_{b,m,n}$ be the r.v for the number of cars that overflow from a hash table of length m and size n with buckets of size b. Then*

$$\begin{aligned}
\mathbf{E}[\Omega_{b,m,n}] &= \sum_{i=2}^{\lfloor n/b \rfloor} \binom{n}{i} \frac{(-1)^i}{m^i} \sum_{k=1}^{m} k^{i-1} \binom{bk-i}{bk-1}, \\
\mathbf{E}[\Omega_{b,m,bm-1}] &= \frac{\sqrt{2\pi bm}}{4} - \frac{7}{6} + \sum_{d=1}^{b-1} \frac{1}{1 - T\left(e^{\frac{2\pi i d}{b}-1}\right)} + \frac{1}{48}\sqrt{\frac{2\pi}{bm}} + O\left(\frac{1}{bm}\right).
\end{aligned}$$

It is important to notice that these results have already been presented in [49]. For $b = 1$ and after using the identity

$$\sum_{k=0}^{r} \binom{r}{k} k^r = (-1)^r r!,$$

we rederive the known result presented in [21]

$$\mathbf{E}[\Omega_{1,m,n}] = \frac{1}{2}\left(Q_0(m,n) - 1 - \frac{n}{m}\right),$$

where $Q_0(m,n) = \sum_{i\geq 0} \frac{n^{\underline{i}}}{m^i}$ is the Ramanujan Q function.

9 Analysis of Robin Hood Linear Probing

In this section we find the distribution of the cost of a successful search for a random element in a hash table of size $m \to \infty$ that contains $n = b\alpha m$ elements with $0 \leq \alpha < 1$.

Let $\Psi_m(b\alpha, z)$ be the probability generating function for the cost of a successful search for a random element in a $b\alpha$-full table with m buckets of size b and $\alpha < 1$.

As mentioned in Section 5 the cost of retrieving a random element is composed by all the elements that hash to the same location (collisions), plus the number of elements that overflow from the previous location. We first derive the generating function $C_m(b\alpha, z)$ for the total displacement, that is, the generating function for the total number of comparisons, without considering the fact that we have to count only the number of buckets probed. Then, if

$$C_m(b\alpha, z) = \sum_{i\geq 0} c_{m,i}(b\alpha) z^i,$$

the probability generating function for the cost of a successful search is

$$\begin{aligned}
\Psi_m(b\alpha, z) &= z\sum_{i\geq 0} c_{m,i}(b\alpha) z^{\lfloor \frac{i}{b} \rfloor} = z\sum_{k\geq 0}\left(\sum_{d=0}^{b-1} c_{m,bk+d}(b\alpha)\right) z^k \\
&= \frac{z}{b}\sum_{d=0}^{b-1} C_m\left(b\alpha, r^d z^{1/b}\right) \sum_{p=0}^{b-1}\left(r^d z^{1/b}\right)^{-p}, \qquad (32)
\end{aligned}$$

where $r = e^{\frac{2\pi i}{b}}$ is a b-th root of unity. Since the calculations are very involved we extract exact coefficients when possible, and then use the Poisson Transform to find the results in the exact model. A similar approach is used to find higher moments.

If k elements collide with the searched one, the expected total displacement originated by these collisions for (separately) retrieving all these elements. is

$$\frac{1}{k+1}\sum_{r=0}^{k} z^r = \frac{1}{k+1}\left(\frac{1-z^{k+1}}{1-z}\right).$$

Since the probability of having k elements. colliding with the searched one is $e^{-b\alpha}\frac{(b\alpha)^k}{k!}$, we immediately see that the probability generating function of the displacement originated by these collisions is

$$\frac{e^{-b\alpha}}{1-z}\sum_{k\geq 0}\frac{(b\alpha)^k}{(k+1)!}\left(1-z^{k+1}\right) = \frac{1-e^{b\alpha(z-1)}}{b\alpha(1-z)}. \tag{33}$$

To conclude the derivation we have to consider the cost originated by the elements that overflow, and so, by Theorem 9 and equation (33) we find

$$C_m(b\alpha, z) = \frac{b(1-\alpha)(1-e^{b\alpha(z-1)})}{b\alpha\left(z^b - e^{b\alpha(z-1)}\right)}\frac{\prod_{j=1}^{b-1}\left(z - \frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}{\prod_{j=1}^{b-1}\left(1 - \frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)} + O\left(\alpha^{bm}\right). \tag{34}$$

When $b=1$, $T_0(\alpha) = (1-\alpha)$, and so we obtain

$$\lim_{m\to\infty} C_m(\alpha, z) = \frac{1-\alpha}{\alpha}\frac{1-e^{\alpha(z-1)}}{z-e^{\alpha(z-1)}},$$

as derived in [48] and [25].

To extract the coefficients $\psi_k(b\alpha)$ we first find $c_n(b\alpha)$. Towards this goal we expand equation (34)

$$\begin{aligned}
C_m(b\alpha, z) &= \left(1-z^{-b}\right)\left(\sum_{s=0}^{b-1}\frac{T_{b-1-s}(b\alpha)}{b\alpha}z^s\right)\sum_{k\geq 1} e^{-kb\alpha}\frac{e^{kb\alpha z}}{z^{bk}} + O\left(\alpha^{bm}\right) \\
&= \left(1-z^{-b}\right)\left(\sum_{s=0}^{b-1}\frac{T_{b-1-s}(b\alpha)}{b\alpha}z^s\right)\sum_{n\geq 0}\left(\sum_{k\geq 1} e^{-kb\alpha}\frac{(kb\alpha)^{n+bk}}{(n+bk)!}\right) z^n + O\left(\alpha^{bm}\right) \\
&= \sum_{n\geq 0}\sum_{s=0}^{b-1}\frac{T_{b-1-s}(b\alpha)}{b\alpha}\sum_{k\geq 1} e^{-kb\alpha}\left(\frac{(kb\alpha)^{n-s+bk}}{(n-s+bk)!} - \frac{(kb\alpha)^{n-s+b(k+1)}}{(n-s+b(k+1))!}\right) z^n + O\left(\alpha^{bm}\right)
\end{aligned} \tag{35}$$

As a consequence, from equations (32) and (35) we have proved the following theorem

Theorem 13. *Let $\Psi_{m,b\alpha}$ be the random variable for the cost of searching a random element in a $b\alpha$-full table with m buckets of size b and $\alpha < 1$, using the Robin Hood linear probing hashing algorithm, and let $\Psi_m(b\alpha, z)$ be its probability generating function. Then*

$$\begin{aligned}
\Psi_m(b\alpha, z) &= \frac{z}{b}\sum_{d=0}^{b-1} C_m\left(b\alpha, e^{\frac{2\pi \mathrm{i} d}{b}} z^{1/b}\right)\sum_{p=0}^{b-1}\left(e^{\frac{2\pi \mathrm{i} d}{b}} z^{1/b}\right)^{-p}, \quad \text{with} \\
C_m(b\alpha, z) &= \frac{1-e^{b\alpha(z-1)}}{b\alpha\left(z^b - e^{b\alpha(z-1)}\right)}\sum_{s=0}^{b-1} T_{b-1-s}(b\alpha) z^s + O\left(\alpha^{bm}\right) \\
&= \frac{b(1-\alpha)(1-e^{b\alpha(z-1)})}{b\alpha\left(z^b - e^{b\alpha(z-1)}\right)} \frac{\prod_{j=1}^{b-1}\left(z - \frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)}{\prod_{j=1}^{b-1}\left(1 - \frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)} + O\left(\alpha^{bm}\right).
\end{aligned}$$

Moreover, the probability $\psi_i(b\alpha)$ that $i+1$ buckets have to be probed to retrieve a random element is

$$\begin{aligned}
\psi_i(b\alpha) \quad &= \quad Pr\{\Psi_{b\alpha} = i+1\} = \\
\sum_{s=0}^{b-1} \frac{T_{b-1-s}(b\alpha)}{b\alpha} &\sum_{k\geq 1} e^{-kb\alpha} \sum_{d=0}^{b-1}\left(\frac{(kb\alpha)^{b(i+k)+d-s}}{(b(i+k)+d-s)!} - \frac{(kb\alpha)^{b(i+k+1)+d-s}}{(b(i+k+1)+d-s)!}\right) + O\left(\alpha^{bm}\right).
\end{aligned} \tag{36}$$

Even though the calculations are very involved and the expressions very complicated, we are able to present amazingly simple expressions for the first moment, in both models. First, the expected value is obtained in the Poisson model, and then by depoissonization the respective result is obtained in the exact model

Theorem 14. *Let $\Psi_{b,m,n}$ be the random variable for the cost of searching a random element when we insert $n+1$ elements in a hash table of m buckets of size b using the Robin Hood linear probing hashing algorithm. Then for $0 \leq \alpha < 1$ we have*

$$\mathbf{E}[\Psi_{b,m,n+1}] = 1 + \sum_{k=1}^{\lfloor n/b \rfloor}\sum_{i=kb}^{n}(-1)^{i-kb}\binom{i-1}{kb-1}\frac{(kb)^i}{(i+1)!}\frac{n^{\underline{i}}}{(bm)^i}, \tag{37}$$

$$\mathbf{E}[\Psi_{m,b\alpha}] = 1 + \sum_{k\geq 1} e^{-kb\alpha}\sum_{n\geq 1} n\frac{(kb\alpha)^{n+bk-1}}{(n+bk)!} + O\left(\alpha^{bm}\right), \tag{38}$$

$$\mathbf{E}[\Psi_{m,b\alpha}] = 1 + \frac{1}{b\alpha}\left(\frac{1}{2}\left(\frac{1}{1-\alpha} - b(1-\alpha)\right) + \sum_{d=1}^{b-1}\frac{1}{\left(1 - \frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)}\right) + O\left(\alpha^{bm}\right), \tag{39}$$

$$b\mathbf{E}[\Psi_{b,m,bm-1}] = \frac{\sqrt{2\pi bm}}{4} + \frac{1}{3} + \sum_{d=1}^{b-1}\frac{1}{1 - T\left(e^{\frac{2\pi \mathrm{i} d}{b} - 1}\right)} + \frac{1}{48}\sqrt{\frac{2\pi}{bm}} + O\left(\frac{1}{bm}\right), \tag{40}$$

where $T(u)$ is the Tree Function ($T(u) = ue^{T(u)}$).

Proof: While equation (37) is a new result, the special case for a full table ($n = bm-1$) is derived in [49]. Moreover equation (38) appears in [29].

It is important to notice that there are two equivalent expressions for $\mathbf{E}[\Psi_{m,b\alpha}]$. Equation (38) can be directly derived from the explicit formula for $\psi_i(b\alpha)$ from Theorem 13, equation (39) is derived directly from the generating function, and uses the properties of the Tuba Numbers. It is important to notice that if we expand the generating functions in the sum of equation (39) we obtain directly equation (38). Let

$$A(b\alpha, z) = \frac{1-e^{b\alpha(z-1)}}{b\alpha\left(z^b - e^{b\alpha(z-1)}\right)}, \qquad B(b\alpha, z) = b(1-\alpha)\frac{\prod_{j=1}^{b-1}\left(z - \frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}{\prod_{j=1}^{b-1}\left(1 - \frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}.$$

When we differentiate $\Psi_m(b\alpha, z)$, since $\sum_{d=0}^{b-1} r^d = 0$ when $d \neq 0$, then most of the terms are null after using the operator $\mathrm{U_z}$, and so we obtain

$$\begin{aligned} \mathrm{U_z}\partial_z\Psi_m(b\alpha, z) &= \mathrm{U_z}\left(\frac{1}{b}\left(\partial_z A(b\alpha, z)B(b\alpha, z) + A(b\alpha, z)\partial_z B(b\alpha, z) - A(b\alpha, z)B(b\alpha, z)\frac{b-1}{2}\right)\right. \\ &\qquad \left. -\frac{1}{b}\sum_{d=1}^{b-1} A(b\alpha, r^d z)B(b\alpha, r^d z)\frac{1}{b}\sum_{p=0}^{b-1} pr^{-dp}\right) + O\left(\alpha^{bm}\right), \end{aligned} \tag{41}$$

with

$$\mathrm{U_z}A(b\alpha, z) = \frac{1}{b(1-\alpha)}, \qquad \mathrm{U_z}\partial_z A(b\alpha, z) = \frac{1}{2}\left(\frac{1}{b(1-\alpha)^2} - \frac{1}{(1-\alpha)}\right),$$

$$\mathrm{U_z}B(b\alpha, z) = b\alpha, \qquad \mathrm{U_z}\partial_z B(b\alpha, z) = b(1-\alpha)\sum_{d=1}^{b-1}\frac{1}{\left(1 - \frac{T(r^d\alpha e^{-\alpha})}{\alpha}\right)},$$

$$\mathrm{U_z}A(b\alpha, r^d z) = -\frac{1}{b\alpha}, \qquad \mathrm{U_z}B(b\alpha, r^d z) = \sum_{s=0}^{b-1} T_{b-1-s}(b\alpha)(r^d)^s, \quad 1 \le d \le b-1.$$

Moreover we have the following chain of identities:

$$\begin{aligned} -\frac{1}{b}\sum_{d=1}^{b-1} A(b\alpha, r^d z)B(b\alpha, r^d z)\frac{1}{b}\sum_{p=0}^{b-1} pr^{-dp} &= \frac{1}{b}\sum_{d=1}^{b-1}\frac{1}{b\alpha}\sum_{s=0}^{b-1} T_{b-1-s}(b\alpha)r^{ds}\frac{1}{b}\sum_{p=0}^{b-1} pr^{-dp} \\ &= \frac{1}{b^3\alpha}\sum_{s=0}^{b-1} T_{b-1-s}(b\alpha)\sum_{p=0}^{b-1} p\sum_{d=1}^{b-1}\left(r^{s-p}\right)^d. \end{aligned}$$

Since

$$\sum_{d=1}^{b-1}\left(r^{s-p}\right)^d = \begin{cases} b-1 & \text{if } s = p \\ -1 & \text{otherwise,} \end{cases}$$

then

$$\begin{aligned}
\frac{1}{b^3\alpha}\sum_{s=0}^{b-1}T_{b-1-s}(b\alpha)\sum_{p=0}^{b-1}p\sum_{d=1}^{b-1}\left(r^{s-p}\right)^d &= -\frac{b-1}{2b^2\alpha}\sum_{s=0}^{b-1}T_{b-1-s}(b\alpha)+\frac{1}{b^2\alpha}\sum_{s=0}^{b-1}sT_{b-1-s}(b\alpha)\\
&= -\frac{b-1}{2b^2\alpha}\mathrm{U_z}B(b\alpha,z)+\frac{1}{b^2\alpha}\mathrm{U_z}\partial_z B(b\alpha,z)\\
&= -\frac{(b-1)(1-\alpha)}{2b\alpha}+\frac{1-\alpha}{b\alpha}\sum_{d=1}^{b-1}\frac{1}{\left(1-\frac{T(r^d\alpha e^{-\alpha})}{\alpha}\right)}.
\end{aligned}$$

The result follows after substituting all these identities in equation (41). □

From Theorem 13 we may derive exact expressions for the factorial moments. We present here the main results in the Poisson model, that generalize the results presented in [48] leaving the details of the derivations since they are mechanical manipulation of the generating functions and their derivatives. The corresponding expressions in the exact model can be obtained with the use of the Poisson transform, although they are extremely complicated.

We first obtain the exact expression for all the factorial moments of $\Psi_{m,b\alpha}$. Given the probability generating function $\Psi_m(b\alpha,z)$, then $E[\Psi_{m,b\alpha}^{\underline{r}}]$ can be can be obtained by differentiating this generating function r times and setting $z=1$.

Theorem 15. *Let $\Psi_{m,b\alpha}$ be the random variable for the cost of searching a random element in a $b\alpha$-full table with m buckets of size b and $\alpha<1$, using the Robin Hood linear probing hashing algorithm, and let $\Psi_m(b\alpha,z)$ be its probability generating function. Then*

$$\begin{aligned}
E[\Psi_{m,b\alpha}^{\underline{r}}] &= r\sum_{s=0}^{b-1}\frac{T_{b-1-s}(b\alpha)}{b\alpha}\sum_{n\geq 1}n^{\underline{r-1}}\sum_{k\geq 1}e^{-kb\alpha}\sum_{d=0}^{b-1}\frac{(kb\alpha)^{b(k+n)+d-s}}{(b(k+n)+d-s)!}\\
&+ 1^{\underline{r}}\sum_{s=0}^{b-1}\frac{T_{b-1-s}(b\alpha)}{b\alpha}\sum_{k\geq 1}e^{-kb\alpha}\sum_{d=0}^{b-1}\frac{(kb\alpha)^{bk+d-s}}{(bk+d-s)!}\\
&= r\frac{1-\alpha}{\alpha}\sum_{n\geq 1}(n+1)^{\underline{r-1}}\sum_{k\geq 1}e^{-kb\alpha}\sum_{d=0}^{b-1}\frac{(kb\alpha)^{b(k+n)+d}}{(b(k+n)+d)!}\\
&- r(r-1)\sum_{n\geq 1}n^{\underline{r-2}}\sum_{k\geq 1}e^{-kb\alpha}\sum_{d=0}^{b-1}\frac{(kb\alpha)^{b(k+n)+d}}{(b(k+n)+d)!}\sum_{s=0}^{b-1-d}\frac{T_{b-1-s}(b\alpha)}{b\alpha}\\
&+ 1^{\underline{r-1}}r\sum_{d=0}^{b-1}\sum_{k\geq 1}e^{-kb\alpha}\frac{(kb\alpha)^{bk+d}}{(bk+d)!}\sum_{s=b-d}^{b-1}\frac{T_{b-1-s}(b\alpha)}{b\alpha}\\
&+ 1^{\underline{r}}\sum_{s=0}^{b-1}\frac{T_{b-1-s}(b\alpha)}{b\alpha}\sum_{k\geq 1}e^{-kb\alpha}\sum_{d=0}^{b-1}\frac{(kb\alpha)^{bk+d-s}}{(bk+d-s)!}+O\left(\alpha^{bm}\right). \qquad (42)
\end{aligned}$$

It is important to notice that the main asymptotic contribution when $\alpha\to 1$ is given by the first sum of equation (42), while the last two sums vanish for moments greater than 2.

Acknowledgements

The author wants to thank Patricio Poblete and an anonymous referee for their very useful comments in earlier versions of the paper, that lead to important improvements in its presentation.

References

[1] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions*. Dover Publications, Inc., New York, 1972.

[2] O. Amble and D. E. Knuth. Ordered hash tables. *Computer Journal*, 17(2):135–142, 1974.

[3] Bruce C. Berndt. *Ramanujan's Notebooks, Part II*. Springer Verlag, 1989.

[4] I.F. Blake and A.G. Konheim. Big buckets are (are not) better! *J. ACM*, 24(4):591–606, October 1977.

[5] R.P. Brent. Reducing the retrieval time of scatter storage techniques. *C. ACM*, 16(2):105–109, 1973.

[6] A. Broder. Two counting problems solved via string encodings. In A. Apostolico and Z. Galil, editors, *Combinatorial Algorithms on Words*, volume 12 of NATO Advance Science Institute Series. Series F: Computer and System Sciences, pages 229–240. Springer Verlag, 1985.

[7] S. Carlsson, J.I. Munro, and P.V. Poblete. On linear probing hashing. Unpublished Manuscript, 1987.

[8] P. Celis. *Robin Hood Hashing*. PhD thesis, Computer Science Department, University of Waterloo, April 1986. Technical Report CS-86-14.

[9] P. Celis, P.-Å. Larson, and J.I. Munro. Robin Hood hashing. In *26th IEEE Sympusium on the Foundations of Computer Science*, pages 281–288, 1985.

[10] P. Chassaing and G. Louchard. Phase transition for parking blocks, brownian excursion and coalescence. *Random Structures & Algorithms*, 21(1):76–119, 2002.

[11] P. Chassaing and J.-F. Marckert. Parking functions, empirical processes, and the width of rooted labeled trees. *Electronic Journal of Combinatorics*, 8(1), 2001.

[12] R. Fagin, J. Nievergelt, N. Pippenger, and H. R. Strong. Extendible hashing - a fast access method for dynamic files. *ACM Transactions on Database Systems*, 4(3):315–344, 1979.

[13] P. Flajolet, X. Gourdon, and P. Dumas. Mellin transforms and asymptotics : Harmonic sums. *Theoretical Computer Science*, 144(1–2):3–58, 1995.

[14] P. Flajolet, P. Grabner, P. Kirschenhofer, and H. Prodinger. On Ramanujan's Q-function. *Journal of Computational and Applied Mathematics*, 58(1):103–116, 1995.

[15] P. Flajolet, P. Poblete, and A. Viola. On the analysis of linear probing hashing. *Algorithmica*, 22(4):490 – 515, 1998.

[16] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

[17] Philippe Flajolet and Andrew M. Odlyzko. Random mapping statistics. In J-J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology*, volume 434 of *Lecture Notes in Computer Science*, pages 329–354. Springer Verlag, 1990. Proceedings of EUROCRYPT'89, Houtalen, Belgium, April 1989.

[18] Philippe Flajolet and Andrew M. Odlyzko. Singularity analysis of generating functions. *SIAM Journal on Discrete Mathematics*, 3(2):216–240, 1990.

[19] G. H. Gonnet and R. Baeza-Yates. *Handbook of Algorithms and Data Structures: in Pascal and C*. Addison–Wesley, second edition, 1991.

[20] G.H. Gonnet and J.I. Munro. Efficient ordering of hash tables. *SIAM Journal on Computing*, 8(3):463–478, 1979.

[21] G.H. Gonnet and J.I. Munro. The analysis of linear probing sort by the use of a new mathematical transform. *Journal of Algorithms*, 5:451–470, 1984.

[22] I. P. Goulden and D. M. Jackson. *Combinatorial Enumeration*. John Wiley, New York, 1983.

[23] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley Publishing Company, 1989.

[24] P. Jacquet and M. Régnier. Trie partitioning process: Limiting distributions. In P. Franchi-Zanetacchi, editor, *CAAP'86*, volume 214 of *LNCS*, pages 196–210, 1986. Proceedings of the 11th Colloquium on Trees in Algebra and Programming, Nice France, March 1986.

[25] Svante Janson. Individual displacements for linear probing hashing with different insertion policies, 2005.

[26] D. E. Knuth. Notes on "open" addressing. Unpublished memorandum, 1963. (Memo dated July 22, 1963. With annotation "*My first analysis of an algorithm, originally done during Summer 1962 in Madison*". Also conjectures the asymptotics of the Q-function, with annotation "*Proved May 24, 1965*".).

[27] D. E. Knuth. Analysis of optimum caching. *Journal of Algorithms*, 6:181–199, 1985.

[28] D. E. Knuth. *The Art of Computer Programming*, volume 1 Fundamental Algorithms. Addison-Wesley Publishing Company, 1997.

[29] D. E. Knuth. *The Art of Computer Programming*, volume 3 Sorting and Searching. Addison-Wesley Publishing Company, 1998.

[30] D. E. Knuth. *The Art of Computer Programming*, volume 2 Seminumerical Algorithms. Addison-Wesley Publishing Company, 1998.

[31] D. E. Knuth and G. S. Rao. Activity in an interleaved memory. *IEEE Transactions on Computers*, C-24:943–944, 1975.

[32] D. E. Knuth and A. Schönhage. The expected linearity of a simple equivalence algorithm. *Theoretical Computer Science*, 6:281–315, 1978.

[33] Donald E. Knuth. Linear probing and graphs. *Algorithmica*, 22(4):561–568, 1998.

[34] A.G. Konheim and B. Weiss. An occupancy discipline and applications. *SIAM Journal on Applied Mathematics*, 6(14):1266–1274, 1966.

[35] P.-Å. Larson. Analysis of uniform hashing. *JACM*, 30(4):805–819, 1983.

[36] H. Mendelson. Analysis of linear probing with buckets. *Information Systems*, 8(3):207–216, 1983.

[37] J. W. Moon. Counting labelled trees. In *Canadian Mathematical Monographs*, volume 1. Canadian Mathematical Congress, 1970.

[38] A. M. Odlyzko. Asymptotic enumeration methods. In M. Grötschel R. Graham and L. Lovász, editors, *Handbook of Combinatorics*, volume II, pages 1063–1229. Elsevier, Amsterdam, 1995.

[39] T. Prellberg P. Cameron, D.Johannsen and P. Schweitzer. Counting defective parking functions. *Electronic Journal of Combinatorics*, 15(1), 2008.

[40] A. Panholzer. On a discrete parking problem. 2008.

[41] W. W. Peterson. Addressing for random-access storage. *IBM Journal of Research and Development*, 1(2):130–146, 1957.

[42] P.V. Poblete. Approximating functions by their Poisson transform. *Information Processing Letters*, 23:127–130, 1986.

[43] P.V. Poblete and J.I. Munro. Last-come-first-served hashing. *Journal of Algorithms*, 10:228–248, 1989.

[44] P.V. Poblete, A. Viola, and J.I. Munro. The Diagonal Poisson Transform and its application to the analysis of a hashing scheme. *Random Structures & Algorithms*, 10(2):221–255, 1997.

[45] Robert Sedgewick. *Algorithms in C, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching*. Addison–Wesley, Reading, Mass., third edition, 1998.

[46] G. Seitz. Parking functions and generalizations. Diploma Thesis, TU Wien, 2009.

[47] A. Viola. *Analysis of Hashing Algorithms and a New Mathematical Transform*. PhD thesis, Computer Science Department, University of Waterloo, November 1995. Technical Report CS-95-50.

[48] A. Viola. Exact distribution of individual displacements in linear probing hashing. *ACM Transactions on Algorithms*, 1(2):214–242, 2005.

[49] A. Viola and P. V. Poblete. The analysis of linear probing hashing with buckets. *Algorithmica*, 21(1):37–71, 1998.

[50] H. S. Wilf. *generatingfunctionology*. Academic Press, 1994.