Dimensioning a multiple hashing scheme

Barbour, A D ; Phatarfod, R

DOI: https://doi.org/10.2307/3215386

Dimensioning a multiple hashing scheme

A. D. Barbour and R. M. Phatarfod

Universität Zürich and Monash University

Version of 31.7.95

Abstract

The number of items of data which are irretrievable without additional effort after hashing can be greatly reduced if several hash tables are used simultaneously. Here we show that, in a multiple hashing scheme, this number has a distribution very close to Poisson. Thus choosing the number and sizes of the tables to minimize the *expected* number of irretrievable items is the right way to dimension a scheme.

Acknowledgement This work was accomplished while ADB was visiting the Mathematics Department at Monash University; their warm hospitality is gratefully acknowledged.

1.a Introduction

A problem of considerable importance in computer science is how to store information so that it can be searched and retrieved efficiently. Typically, information is stored as items of data, and a key is associated with each item. For example, in a university's record of its students, the key could be a student's name and the data his or her address. The problem of storage is tackled in various ways, depending upon the number and nature of the keys involved, and the eventual use of the information retrieved. One way to store information, useful when the subset of all possible keys that actually occur in the data is not known in advance, is with a technique called *hashing;* see Kruse (1987). A pointer to the item of data corresponding to each key is stored in an array called a *hash table,* the position in the array being allocated by applying a *hashing function* to the key as argument; this makes for speedy storage as well as for speedy retrieval.

An error occurs if the hashing function, applied to two different keys, results in the same position in the hash table, and so the ideal hashing function would apparently be one to one from the set of all possible keys to the positions in the hash table. This, however, would usually require the provision of a table of enormous size: for instance, all 'possible' eight letter names from a 26 letter alphabet would require an array of size about 2×10^{11} positions. In practice, a much smaller array can be provided if the hashing function acts on the key much as a random number generator acts on its seed, producing a 'pseudo–random uniformly distributed' position in the array; what is more, such functions can be efficiently implemented. So, if we have m keys and n positions in the array, the process of hashing the keys is similar to that of allocating at random m balls to n boxes. As in the latter case, the probability that any two given keys are hashed to the same position is easily made small by choosing n, the size of the hash table, to be large — in the allocation model, it is $1/n$ — but the probability of there being two or more keys which are hashed to the same position need not be small, even if n is considerably greater than m; as the birthday problem shows, with $m = 23$ balls and $n = 365$ boxes, the probability of a box containing two or more balls is about one half. The event of two or more keys being hashed to the same position is called a *collision.* In practice, there are various techniques for retrieving the desired information when a collision occurs, but the additional effort is such that it is desirable to try to minimize the number of these errors.

Recently, Srinivasan *et al* (1995) suggested a variation of the technique, called Multiple (Bit) Hashing. This differs from the standard procedure by using $k > 1$ independent hashing functions. These hash the keys into k hash tables, with sizes $r_1, \ldots, r_k$ such that $\sum_{l=1}^{k} r_l = n$. As before, each hash function is so chosen that the keys are distributed

randomly within each table. The situation is now similar to that of having m labelled balls being allocated at random to each of k arrays of boxes. If the total number n of boxes is kept fixed, the number of boxes in each of the arrays must be reduced, making the probability of a collision in any particular array much greater. However, the probability of a key having its hashed position in collision in *every* array may now be very much smaller than the probability of collision when hashing into a single hash table. Thus, to be able to retrieve information accurately with low probability of failure, it is enough, when storing each item of information, to hash its key with each of the k hashing functions, and to write the pointer to the item in the corresponding position in each hash table; if a collision occurs in any table, the number of keys currently hashing to that position is written instead of the pointer. An item can then be retrieved, provided that it does not collide in *every* table, and items can also be removed from the data base. Furthermore, as is important for spell checkers or for security screening, a key which does *not* correspond to an item in the data base is readily identified, because it hashes to an empty position in one or more of the hash tables.

There are two ways in which such a system can fail. First, a key which is not in the data base may nonetheless happen to have a collision in every hash table, and may thus falsely be accepted as being present in the data base — the elementary acceptance rule. In the allocation model, hashing new keys corresponds to allocating new balls to boxes in each of the arrays, uniformly and independently of each other and of all else, so that the failure process is modelled by a sequence of independent Bernoulli trials with some probability π of failure: calculation shows that

$$\pi = \prod_{l=1}^{k} q_l, \qquad \text{with} \qquad q_l = \left\{1 - \left(1 - \frac{1}{r_l}\right)^m\right\}. \tag{1.1}$$

Optimizing the choice of k and $r_1, \ldots, r_k$ with respect to this kind of failure thus simply requires π to be kept to a minimum.

Actually, this kind of failure can be further reduced, by checking whether or not all pointers uniquely identified by hashing the key are identical — the refined acceptance rule. If this is done, the failure probability is reduced to

$$\begin{aligned} \tilde{\pi} &= m \prod_{l=1}^{k} \{s_l + m^{-1} t_l\} - (m-1) \prod_{l=1}^{k} s_l \\ &\sim \left\{1 + \sum_{j=1}^{k} t_j / s_j\right\} \prod_{l=1}^{k} s_l, \end{aligned} \tag{1.2}$$

where

$$s_l = q_l - t_l \qquad \text{and} \qquad t_l = \frac{m}{r_l}\Big(1 - \frac{1}{r_l}\Big)^{m-1}, \tag{1.3}$$

and the asymptotics are valid if $m \to \infty$, the t_l/s_l are uniformly bounded and $k/m \to 0$. Note however that, for a spell checker, much storage can be saved by replacing the pointers by the single bit 1 (Srinivasan *et al* 1995), so that the dictionary itself need not be stored. Any misspelt words, which are keys not present in the data base, can be rejected by the elementary rule, but since all 'pointers' are now the same, the refined rule can no longer be implemented.

The second kind of failure is that in which a key in the data base gives a collision in every hash table, so that no pointer to the associated item of information is found, and the item is irretrievable. The probability according to the allocation model of a given item being irretrievable is given by

$$\pi' = \prod_{l=1}^{k} q'_l, \qquad \text{where} \qquad q'_l = \Big\{1 - \Big(1 - \frac{1}{r_l}\Big)^{m-1}\Big\}, \tag{1.4}$$

and so the proportion of irretrievable items is very close to the probability of a key being falsely taken to be present, using the elementary acceptance rule. However, the events that keys are irretrievable are no longer independent; for instance, if $k = 1$, one such event has to entail at least one other. Thus, if k and $r_1, \ldots, r_k$ are to be chosen to make the (random) total number W of irretrievable items as small as possible, there are many different forms of comparison which could be used — smallest expectation $\mathbb{E}W$, smallest probability that W exceeds a given value, smallest weighted average of several such tail probabilities, and so on. Our aim in this paper is to show that, for the current problem, consideration of $\mathbb{E}W = m\pi'$ is enough for all practical purposes. Hence, for large m, it is enough to choose k and $r_1, \ldots, r_k$ so as to minimize π', and since π and π' are almost indistinguishable for large m, this choice also minimizes the proportion of failures of the first kind, under the elementary acceptance rule.

The best choice of k and $r_1, \ldots, r_k$ for these purposes is obtained as follows. The problem is the same as maximizing

$$-\log \pi = -\sum_{l=1}^{k} \log\Big\{1 - \Big(1 - \frac{1}{r_l}\Big)^{m}\Big\}$$

in positive integral k and $r_1, \ldots, r_k$, under the constraint $\sum_{l=1}^{k} r_l = n$. An almost equivalent, continuous version of the problem is to maximize

$$-\int_0^1 x \log\{1 - (1-x)^m\}\mu(dx)$$

over arbitrary non–negative measures μ on $(0,1)$ satisfying $\int_0^1 \mu(dx) = n$, the relation with the original problem following by taking $\mu\{r_l^{-1}\} = r_l$, $1 \le l \le k$. Here, the solution can easily be seen to be for μ to put mass n on the point x_*, where x_* is chosen to maximize $-x\log\{1-(1-x)^m\}$ in $0 < x < 1$. Calculation shows that x_* then satisfies $(1-x_*)^m = \frac{1}{2} + O(m^{-1})$, or $mx_* = \log 2 + O(m^{-1})$. The solution corresponds, in the original notation, to taking all the r_l equal to

$$r_* = x_*^{-1} = \frac{m}{\log 2} + O(1),$$

with $k = k_* = n/r_*$. The constraint that k and the r_l's all have to be integral makes the exact best choice a slight perturbation of this ideal, but the sub–optimal choice of $k = [n/\hat{r}]$, where $\hat{r} = [1/x_*]$ and $[\cdot]$ denotes the integer part, shows that, in any event,

$$\max_{k;r_1,\ldots,r_k} \Big(-\sum_{l=1}^{k} \log\Big\{1-\Big(1-\frac{1}{r_l}\Big)^m\Big\}\Big) = \frac{n}{m}(\log 2)^2\Big\{1+O\Big(\frac{1}{m}+\frac{m}{n}\Big)\Big\},$$

with smallest attainable value of π no larger than

$$\Big\{1-\Big(1-\frac{1}{\hat{r}}\Big)^m\Big\}^{[n/\hat{r}]},$$

of order $O\Big(2^{-(n/m)\log 2}\Big)$ uniformly in $n \le m^2$. This, for example, compares extremely favourably with an expected proportion of failures of order m/n if only a single hash table $(k=1)$ is used, as soon as n/m is at all large.

1.b Main theorem

As before, we describe the hashing scheme by the following allocation model. n boxes are arranged into k rows, with row l containing r_l boxes, and with $\sum_{l=1}^{k} r_l = n$. In each row, each of $m+1$ labelled balls is assigned uniformly at random to one of the boxes, independently of everything else: we now use $m+1$ instead of m, to simplify the formulae a little. For each $1 \le i \le m+1$, set $I_i = 1$ if, in every row, the ball labelled i shares its box with at least one other ball, and let $W = \sum_{i=1}^{m+1} I_i$. The random variable W counts the number of irretrievable items, and is to be kept as small as possible by suitably choosing k and $r_1, \ldots, r_k$. Note that choosing $r_l = 1$ for any l is merely equivalent to reducing n by one, which is of no help, so that we may assume $r_l \ge 2$ for all l from now on.

A natural way to express the typical size of the count W is through its mean $\mathbb{E}W = (m+1)\pi$, where, as before,

$$\pi = \prod_{l=1}^{k} q_l, \quad \text{and} \quad q_l = \Big\{1 - \Big(1 - \frac{1}{r_l}\Big)^m\Big\}.$$

The problem of choosing k and $r_1, \ldots, r_k$ to minimize $\mathbb{E}W$, or equivalently π, was discussed in the previous section. We now show that the same choice effectively makes W as small as possible in terms of all other reasonable measures of smallness. More precisely, we show that, for a wide range of choices of k and $r_1, \ldots, r_k$, including all those which give reasonably small values of $\mathbb{E}W$, the distribution of W is well approximated by the Poisson distribution $\mathrm{Po}\,(\mathbb{E}W)$ with the same mean. Since the Poisson distributions $\mathrm{Po}\,(\lambda)$ are stochastically increasing in their mean λ, picking the smallest possible mean gives the 'smallest' distribution according to all reasonable criteria of smallness. Thus, apart from the (explicitly estimated) small difference between the true distribution of W and the Poisson distribution $\mathrm{Po}\,(\mathbb{E}W)$, all the information about the size of W is contained in its mean $\mathbb{E}W$.

The following approximation theorem actually goes further: it also shows that tail probabilities of the form $\mathbb{P}[W \geq M]$, or indeed any probability from the distribution of W, can be estimated by the corresponding Poisson probability, again up to an explicitly computable maximum possible error.

Theorem 1.1. *Defining the total variation distance d_{TV} between two probability distributions P and Q on $\mathbb{Z}_+$ by*

$$d_{TV}(P,Q) = \sup_{A \subset \mathbb{Z}_+} |P(A) - Q(A)|,$$

we have

$$d_{TV}(\mathcal{L}(W), \mathrm{Po}\,(\lambda)) \leq (1 - e^{-\lambda}) \prod_{l=1}^{k} \Big(q_l + \frac{1}{m}\Big) \Big\{1 + 3 \sum_{s=1}^{k} \frac{m p_s}{q_s^2 (r_s - 1)}\Big\}, \tag{1.3}$$

where $\lambda = \mathbb{E}W$, $p_l = \Big(1 - \frac{1}{r_l}\Big)^m$, $q_l = 1 - p_l$ and $\mathcal{L}(W)$ denotes the distribution of W.

The bound given in (1.3) can often be simplified. For instance, since $q_l + m^{-1} \leq q_l(1 + r_l m^{-2})$, it follows that

$$\prod_{l=1}^{k} \Big(q_l + \frac{1}{m}\Big) \leq e^{n m^{-2}} \pi.$$

This in turn can be used to prove the following corollary to Theorem 1.

Corollary 1.2. *With notation as before,*

$$d_{TV}(\mathcal{L}(W), \mathrm{Po}\,(\lambda)) \le e^{nm^{-2}}(1-e^{-\lambda})\Big[1+\min\Big\{3k, \frac{9n}{2m}\Big\}\Big]\prod_{l=2}^{k} q_{(l)}, \tag{1.4}$$

where $q_{(1)} \le q_{(2)} \le \ldots \le q_{(k)}$.

It is interesting that choosing $k = 1$ (and thus $r_1 = n$) leads to bounds which are not small. This is because any ball sharing a box in row 1 does so with at least one other ball, and *both* contribute to the number of failures W if $k = 1$. Hence, if $n \gg m$, W has a distribution reasonably close to *twice* the (approximately Poisson) distributed number of boxes containing two or more balls: see Barbour, Holst and Janson (1992), Section 6.2. Thus it should not be expected that W is approximately Poisson distributed when $k = 1$. The same is true if row 1 is the only row that plays a real part in retrieval, as when $r_1 = n - 20$ and $r_2 = \cdots = r_{11} = 2$.

On the other hand, for choices close to that with smallest mean, $k \approx (n/m)\log 2$ and $r_l \approx m/\log 2$, $1 \le l \le k$, the error in the Poisson approximation is of order

$$(n/m)e^{nm^{-2}}2^{-(n/m)\log 2} = O\Big(e^{-(n/m)(\log 2)^2(1+o(1))}\Big),$$

small as long as $n \gg m$. Similarly, for any fixed $C > c > 0$, there is an $\alpha = \alpha(C,c) > 0$ such that the approximation error is of order $O\Big((n/m)e^{-\alpha n/m}\Big)$, uniformly for all choices of k and $r_1, \ldots, r_k$ such that $cm \le r_l \le Cm$, $1 \le l \le k$. Even more generally, if $2m \le n \le m^2$, it follows that $q_{(1)} \ge 3m/4n$, and hence the corollary implies a rough and ready bound of

$$d_{TV}(\mathcal{L}(W), \mathrm{Po}\,(\lambda)) \le 20(n/m)^2\lambda/m. \tag{1.5}$$

This is clearly small as long as n/m is large and $\lambda/m = o((m/n)^2)$; since the optimal value of λ/m is of order $2^{-(n/m)\log 2}$, any remotely reasonable choices of k and $r_1, \ldots, r_k$ for the hashing scheme lead to a good Poisson approximation to the distribution of the number of irretrievable items.

2. Details

As a preliminary to the proofs of Theorem 1 and its corollary, we first need two lemmas.

Lemma 2.1. *For $r \geq 2$, let $p = \left(1 - \frac{1}{r}\right)^m$, $q = 1 - p$. Then*
(i) $\frac{mp}{q(r-1)} \leq 1$;
(ii) $\frac{mp}{qr} \leq \frac{me^{-m/r}}{r(1-e^{-m/r})}$.

Proof. For part (i), observe that

$$\frac{q}{p} = \left(\frac{r}{r-1}\right)^m - 1 = \int_0^{1/(r-1)} m(1+x)^{m-1}\, dx \geq \frac{m}{r-1}.$$

For part (ii), we have $p = \left(1 - \frac{1}{r}\right)^m \leq e^{-m/r}$, and hence

$$p/q = p(1-p)^{-1} \leq e^{-m/r}/(1 - e^{-m/r}).$$

Lemma 2.2. *Let S be a random variable with the binomial* $\mathrm{Bi}\,(m, 1/r)$ *distribution, conditioned to be positive:*

$$\mathbb{P}[S = s] = \binom{m}{s}\left(\frac{1}{r}\right)^s\left(1 - \frac{1}{r}\right)^{m-s} \Big/ \sum_{l=1}^{m}\binom{m}{l}\left(\frac{1}{r}\right)^l\left(1 - \frac{1}{r}\right)^{m-l},$$

for $1 \leq s \leq m$. Then

$$\mathbb{E}(z^S) = \left(1 + \frac{z-1}{r}\right)^m\left\{1 - \left(\frac{r-1}{r-1+z}\right)^m\right\}\Big/\left\{1 - \left(1 - \frac{1}{r}\right)^m\right\}$$

and

$$\mathbb{E}(Sz^S) = \left(\frac{mz}{r-1+z}\right)\left(1 + \frac{z-1}{r}\right)^m\Big/\left\{1 - \left(1 - \frac{1}{r}\right)^m\right\}.$$

In particular, for $z = 1$, we have

$$\mathbb{E}S = \left(\frac{m}{r}\right)\Big/\left\{1 - \left(1 - \frac{1}{r}\right)^m\right\};$$

for $z = \left(1 - \frac{1}{r-1}\right)^{-1}$, we get

$$\mathbb{E}(z^S) = \left\{1 + \frac{1}{r(r-2)}\right\}\left\{1 - \left(1 - \frac{1}{r-1}\right)^m\right\}\Big/\left\{1 - \left(1 - \frac{1}{r}\right)^m\right\}$$

and

$$\mathbb{E}(Sz^S) = \Big(\frac{m}{r-1}\Big)\Big\{1 + \frac{1}{r(r-2)}\Big\}\Big/\Big\{1 - \Big(1 - \frac{1}{r}\Big)^m\Big\}.$$

Proof. Direct calculation.

Turning now to the proof of Theorem 1.1, we appeal to Theorem 1.B of Barbour, Holst and Janson (1992). In our setting, with $\lambda = \mathbb{E}W$, it implies that

$$d_{TV}(\mathcal{L}(W), \mathrm{Po}\,(\lambda)) \le (1 - e^{-\lambda})\mathbb{E}|W - \widetilde{W}|, \tag{2.1}$$

where $W = \sum_{i=1}^{m+1} I_i$ is realized as before, and $\widetilde{W} = \sum_{i=1}^{m} \widetilde{I}_i$: here, $\widetilde{I}_1, \ldots, \widetilde{I}_m$ are any random variables on the same (possibly enlarged) probability space as the I_i, constructed in such a way that

$$\mathcal{L}(\widetilde{I}_1, \ldots, \widetilde{I}_m) = \mathcal{L}(I_1, \ldots, I_m \,|\, I_{m+1} = 1).$$

To construct $\widetilde{I}_i$'s for which $\mathbb{E}|W - \widetilde{W}|$ is small, we proceed as follows.

For $1 \le l \le k$, let L_l take the value 1 if ball $(m+1)$ is alone in row l, zero if not. If $L_l = 0$, let $S_l \ge 1$ denote the number of companions of ball $(m+1)$ in row l; if $L_l = 1$, let $S_l \ge 1$ be sampled, independently of all else, from the binomial distribution $\mathrm{Bi}\,(m, 1/r_l)$ conditioned to take a value greater than or equal to 1:

$$\mathbb{P}[S_l = s] = \binom{m}{s}\Big(\frac{1}{r_l}\Big)^s\Big(1 - \frac{1}{r_l}\Big)^{m-s} q_l^{-1}, \qquad 1 \le s \le m, \tag{2.2}$$

where, as before,

$$q_l = 1 - p_l = \Big\{1 - \Big(1 - \frac{1}{r_l}\Big)^m\Big\}: \tag{2.3}$$

note that $p_l = \mathbb{P}[L_l = 1]$. Then the $(S_l,\ 1 \le l \le k)$ are independent and distributed as in (2.2), and are independent of $(L_l,\ 1 \le l \le k)$. In each row l with $L_l = 1$, S_l balls from the set $\{1, 2, \ldots, m\}$ are taken at random from their original boxes, and are replaced as companions of ball $(m+1)$. The resulting configuration is then a realization from the conditional distribution of the balls, given $I_{m+1} = 1$, and $\widetilde{I}_i = 1$ if ball i now has a companion in every row, $1 \le i \le m$.

To use (2.1), we need to bound $\mathbb{E}|W - \widetilde{W}|$. For this, we observe that

$$\begin{aligned}
\mathbb{E}|W - \widetilde{W}| &\le \mathbb{E}I_{m+1} + \sum_{i=1}^{m} \mathbb{E}|I_i - \widetilde{I}_i| \\
&= \pi + m\{\mathbb{E}(I_1 - \widetilde{I}_1)^+ - \mathbb{E}(I_1 - \widetilde{I}_1)^-\} = \pi + m\{2\mathbb{E}(I_1 - \widetilde{I}_1)^+ - \mathbb{E}(I_1 - \widetilde{I}_1)\} \\
&= \pi + m\{2\mathbb{P}[I_1 = 1 \text{ and } \widetilde{I}_1 = 0] + \mathbb{E}(\widetilde{I}_1 - I_1)\}.
\end{aligned} \tag{2.4}$$

Take first $\mathbb{P}[I_1 = 1 \text{ and } \widetilde{I}_1 = 0]$. If $I_1 = 1$, the event $\widetilde{I}_1 = 0$ occurs if, for at least one l with $L_l = 1$, all the companions of ball 1 are among the S_l balls replaced as new companions of ball $(m+1)$. It thus follows that

$$\begin{aligned} \mathbb{P}[I_1 = 1 \text{ and } \widetilde{I}_1 = 0] &\le \pi \sum_{l=1}^{k} \mathbb{E}(L_l S_l/m) \\ &= \pi \sum_{l=1}^{k} p_l \mathbb{E}(S_l/m) = \pi \sum_{l=1}^{k} p_l/(r_l q_l), \end{aligned} \tag{2.5}$$

from Lemma 2.2. On the other hand, $\widetilde{I}_1 = 1$ if, in each row, after the redistribution of balls, ball 1 has at least one companion, which may possibly be $(m+1)$ itself. This leads to the expression

$$\begin{aligned} \mathbb{E}\widetilde{I}_1 &= \mathbb{E}(I_1 \mid I_{m+1} = 1) \\ &= \prod_{l=1}^{k} \mathbb{E}\Big\{ \Big[1 - \Big(1 - \frac{1}{r_l - 1}\Big)^{m - S_l}\Big] \Big(1 - \frac{S_l}{m}\Big) + \frac{S_l}{m} \Big\} \\ &= \prod_{l=1}^{k} \mathbb{E}\Big\{ \Big[1 - \Big(1 - \frac{1}{r_l - 1}\Big)^{m - S_l}\Big] + \frac{S_l}{m} \Big(1 - \frac{1}{r_l - 1}\Big)^{m - S_l} \Big\}. \end{aligned}$$

Applying Lemma 2.2, we thus find that

$$\begin{aligned} \mathbb{E}\widetilde{I}_1 &= \prod_{l=1}^{k} \Big\{ 1 - \Big(1 - \frac{1}{r_l - 1}\Big)^{m} \Big\{ 1 + \frac{1}{r_l(r_l - 2)} \Big\}^{m} \Big[\Big\{ 1 - \Big(1 - \frac{1}{r_l - 1}\Big)^{m} \Big\} - \frac{1}{r_l - 1} \Big] q_l^{-1} \Big\} \\ &= \prod_{l=1}^{k} \Big\{ 1 - p_l q_l^{-1} \Big[\Big\{ 1 - \Big(1 - \frac{1}{r_l - 1}\Big)^{m} \Big\} - \frac{1}{r_l - 1} \Big] \Big\} \\ &\le \prod_{l=1}^{k} \Big\{ 1 - p_l + \frac{p_l}{q_l(r_l - 1)} \Big\}. \end{aligned}$$

Hence

$$\mathbb{E}(\widetilde{I}_1 - I_1) \le \pi \Big\{ \prod_{l=1}^{k} \Big(1 + \frac{p_l}{q_l^2(r_l - 1)} \Big) - 1 \Big\}. \tag{2.6}$$

Collecting (2.4) – (2.6), we thus have the estimate

$$\mathbb{E}|W - \widetilde{W}| \le \pi \Big\{ 1 + 2 \sum_{l=1}^{k} \frac{m p_l}{r_l q_l} + m \Big[\prod_{l=1}^{k} \Big(1 + \frac{p_l}{q_l^2(r_l - 1)} \Big) - 1 \Big] \Big\}. \tag{2.7}$$

For the statement of Theorem 1.1, we use a simplified but less precise expression. Applying Lemma 1(i) and standard inequalities, we obtain

$$\begin{aligned}\mathbb{E}|W-\widetilde{W}| &\le \prod_{l=1}^{k}(q_l+m^{-1})\Big\{1+2\sum_{l=1}^{k}\frac{mp_l}{r_lq_l}+m\Big[1-\prod_{l=1}^{k}\Big(1+\frac{p_l}{q_l^2(r_l-1)}\Big)^{-1}\Big]\Big\}\\ &\le \prod_{l=1}^{k}(q_l+m^{-1})\Big\{1+2\sum_{l=1}^{k}\frac{mp_l}{r_lq_l}+\sum_{l=1}^{k}\frac{mp_l}{q_l^2(r_l-1)}\Big\}\\ &\le \prod_{l=1}^{k}(q_l+m^{-1})\Big\{1+3\sum_{l=1}^{k}\frac{mp_l}{q_l^2(r_l-1)}\Big\},\end{aligned}$$

and the theorem follows.

To prove Corollary 1.2, as already observed, we have

$$\prod_{l=1}^{k}(q_l+m^{-1})\le e^{nm^{-2}}\prod_{l=1}^{k}q_{(l)},$$

where $q_{(1)}\le\ldots\le q_{(k)}$ denote the ordered q_l's. Then, using Lemma 1(i) and (ii), we note that

$$\frac{mp_l}{q_l^2(r_l-1)}\le\frac{1}{q_{(1)}},$$

giving the estimate

$$\sum_{l=1}^{k}\frac{mp_l}{q_l^2(r_l-1)}\le\frac{k}{q_{(1)}};$$

and also that

$$\frac{mp_l}{q_l^2(r_l-1)}\le\frac{2mp_l}{q_{(1)}q_lr_l}\le\frac{2}{q_{(1)}}\Big\{\frac{me^{-m/r_l}}{r_l(1-e^{-m/r_l})}\Big\},$$

leading to the estimate

$$\begin{aligned}\sum_{l=1}^{k}\frac{mp_l}{q_l^2(r_l-1)} &\le \frac{2}{q_{(1)}}\sum_{l=1}^{k}\Big\{\frac{me^{-m/r_l}}{r_l(1-e^{-m/r_l})}\Big\}\\ &\le \frac{2n}{mq_{(1)}}\sup_{x>0}\Big\{\frac{x^2e^{-x}}{1-e^{-x}}\Big\}<\frac{3n}{2mq_{(1)}}\end{aligned}$$

whenever $\sum_{l=1}^{k}r_l=n$. The corollary now follows.

References

[1] A. D. Barbour, L. Holst and S. Janson (1992) *Poisson approximation.* Clarendon Press, Oxford.

[2] R. L. Kruse (1987) *Data structures and program design,* 2nd Edn. Prentice Hall, New Jersey.

[3] B. Srinivasan, S. Kulkarni and R. M. Phatarfod (1995) A storage efficient structure for dictionary coding. Tech. Rept 95/05, Dept Computer Technology, Monash University, Victoria, Australia.

Postal address Department of Mathematics, Monash University, CLAYTON Vic 3168, Australia.