On the stationary search cost for the Move-To-Root rule with random weights

Javiera Barrera and Christian Paroissin*

CMM - UMR 2071 UCHILE-CNRS
Universidad de Chile
Casilla 170-3, Correo 7
Santiago, Chile
jbarrera@dim.uchile.cl

MODAL'X
Université Paris X Nanterre
200 avenue de la République
92001 Nanterre Cedex, France
cparoiss@u-paris10.fr

Running title: Search cost for the move-to-root rule with random weights.

AMS 2000 Classification: 68W40, 68P10, 44A10.

Keywords: Move-To-Root, Binary search tree, Random discrete distribution, Entropy.

Abstract

Consider a binary search tree containing n items. This tree is updated according to the move-to-root rule as defined first by Allen and Munro [1]. We assume that files have iid random weights, which are used to construct request probabilities. Exact formulas for the two first moments of the stationary search cost are derived from classical result. A formula for the expectation is also given in the case of independent weights. A bound for the expected search cost is obtained. The expected search cost is compared to the one in the case of a list updated according to the move-to-front rule [2]. Classical examples are given to illustrate these results. As a conclusion, we give a brief discussion of our results and some questions are pointed out.

1 Introduction and model

There exists many way to record items into data structures: lists and trees are the two well-known examples. Here we will focus on a special class of trees, called binary search trees. Let us recall few definitions and properties about this data structure. A tree is an acyclic connected graph where the vertices are called nodes. A rooted tree is a directed graph and there exists a unique path from the root to any node j. Every node $i \neq j$ on this path is called an ancestor of j, the closest ancestor being the parent node. The subtree with root i consists of i and all its descendants. A binary tree is an ordered tree in which each node has at most two children (on the left or/and on the right). A node without child is a leaf or a terminal node. A binary search tree is a rooted binary tree in which all items with smaller key values are stored at nodes in the left subtree, while larger or equal key values are stored on the right subtree. Thus, for a given sequence of items, we have the following algorithm to construct a binary search tree ([15], p. 422-451):

Algorithm 1.1 1. If there is not root, insert the item as the root;

2. If the key value is lower than that of the root, insert the item in the left subtree;

*This research has been started while CP was a postdoc at the CMM, Chile.

3. If the key value is greater than that of the root, insert the item in the right subtree;

Notice that the construction of binary search tree depends on the key values of the items but also on the order in which the items are inserted.

Consider n items which are inserted in a binary search tree. This binary search tree is then updated as follows: at each unit of discrete time, an item is requested independently of the previous requests and is moved to the root of the binary tree. This way to update a list is called the move-to-root rule and was suggested by Allen and Munro in [1] (for a review on the move-to-root rule and its generalisations, see the introduction of Bodell's thesis [4]). There exists other methods to maintain a self-adjusting binary search tree: Allen and Munro in [1] also studied the simple exchange (see below); Sleator and Tarjan [19] introduced splay trees and different techniques such like top-down splaying and bottom-up splaying.

The move-to-root rule can be defined as a series of simple exchanges between nodes, performed until the request item has been moved to the the root of the tree. Of course, these exchanges have to preserve the property the data structure of being a binary search tree. The algorithm is the following.

Algorithm 1.2 Let a be the requested item:

1. If a is the root, do nothing;
2. If a is the left child of its parent r, modify the subtree whose root is r as follows:
   - rotate a up to r so that a becomes the root of the subtree;
   - the old left subtree of a is left unchanged in relation to a;
   - the old right subtree of a becomes the left subtree of r
   - the old right subtree of r keeps in relation to r.
3. If a is the right child of its parent r, modify the subtree whose root is r with an analogous transformation.

The aim of this heuristic is to keep a binary search tree in near-optimal form. The associated Markov chain was studied by Dobrow and Fill in [8, 9]. They also proved the existence of a link between this Markov chain and the heaps process (which is the name given to the Markov chain corresponding to the move-to-front rule [10]). Dobrow extended some results to the case of Markovian request [7].

In order to avoid worst cases, one can imagine to combine this heuristic with a balanced tree procedure such like the so-called AVL trees (for which the absolute value of the difference between the depth of any two nodes is zero or one; see [15], p.451-471). Such procedures could be done at each discrete times or when the Markov chain has reach the equilibrium. However, intuitively the probability of failing into a worst case should be small. A numerical evaluation of AVL procedure and self-adjusting binary search trees (like move-to-root) was proposed by Bell and Gupta [3].

Here we consider a binary search tree containing a countable number of items. One can be interested in the search cost of an item in the binary tree at a given time. The search cost is simply defined as the number of ancestors of the item requested (or equivalently as the depth of the item in the tree minus one).

Let $\omega = (w_i)_{i \in I\!N^*}$ be a sequence of independent strictly positive random variables. For any $i \geqslant 1$, w_i represents the weight of the item i. Let us consider the n first files. One can construct a vector of requested probabilities $\mathbf{p_n} = (p_1, \ldots, p_n)$ from weights as follows:

$$\forall i \in \{1, \ldots, n\} \ , \quad p_i = \frac{w_i}{W_n} \qquad \text{where} \qquad W_n = \sum_{i=1}^{n} w_i \ .$$

Such random variables are related to random discrete distributions which are used in many other areas such as Bayesian statistics, combinatorics, genetics, ecology, and analysis of stochastic process (see [13] or [17] for references). Our construction of the move-to-root heuristic with random weights is analogous to the one made in [2] for the move-to-front heuristic with random weights or in [16] for another classical problem, the random coupon collector problem (see also [12]).

We assume that all random variables of the sequence ω have a density function. For any $i \in I\!N^*$, let us denote by f_i the density function of the random variable w_i and by ϕ_i its Laplace transform:

$$\forall t \geqslant 0 \ , \quad \phi_i(t) = I\!E[e^{-w_i t}] = \int_0^\infty e^{-xt} f_i(x) \, dx \ .$$

Let us recall that the ϕ_i's are decreasing functions with $\phi_i(0) = 1$ and tending to 0 as t tends to infinity. We denote by μ_i the expectation of w_i for any $i \in \mathbb{N}^*$: $\mu_i = -\phi_i'(0)$. In the case of a sequence of iid random weights we will forget the subscript: for any $i \in \mathbb{N}^*$, $\phi_i = \phi$ and $\mu_i = \mu$.

In section 2 we give integral representation of the two first moment of the stationary search cost (theorem 2.1 and theorem 2.2) in case of iid weights. The result for the expectation is extended in the case of independent weights. We give a bound for the expectation which is valid for sequence of independent weights. Section 3 is devoted to study three examples. In section 4 we discuss briefly about the results obtained here and two (unanswered) questions which arise naturally at the sight of the results obtained in the two previous section are given.

In this problem, we can also consider a random number of items. To obtain similar results than those shown here, one can consider our problem conditionally to $N = n$ where N is some independent discrete random variable indicating the random number of items.

2 Two first moments of the stationary search cost

In this section, we derive from the results of Allen and Munro [1] the two first moments of the stationary search cost when the items have iid random weights. Let us denote by S_n the stationary search cost.

The following theorem gives an integral representation of the expectation of the stationary search cost S_n:

Theorem 2.1 *For a sequence ω of iid random weights, we have:*

$$\mathbb{E}[S_n] = 2\sum_{i=1}^{n-1} \int_0^\infty \int_t^\infty (n-i)\phi'(u)^2 \phi(u)^{i-1} \phi(t)^{n-i-1}\, du dt\ . \tag{2.1}$$

Proof From theorem 3.1 in [1] (see also [4] where this result is obtained as the limit of the transient expected search cost), we have:

$$\begin{aligned}
\mathbb{E}[S_n] &= \mathbb{E}[\mathbb{E}[S_n \mid \omega]] \\
&= 2\sum_{1 \leqslant i < j \leqslant n} \mathbb{E}\left[\frac{p_i p_j}{p_i + \cdots + p_j}\right] \\
&= 2\sum_{i=1}^{n-1} (n-i)\mathbb{E}\left[\frac{p_1 p_{i+1}}{p_1 + \cdots + p_{i+1}}\right] \\
&= 2\sum_{i=1}^{n-1} (n-i)\mathbb{E}\left[\frac{w_1 w_{i+1}}{(w_1 + \cdots + w_{i+1})W_n}\right]\ ,
\end{aligned}$$

since the weights are identically distributed. Let us denote by E_i the expectation in the summation. Let $i \in \{1, \ldots, n-1\}$. Set $W_n' = w_2 + \cdots + w_i$ and $W_n'' = W_n - W_n' - w_1 - w_{i+1}$. Thus E_i can be expressed as a function of W_n', W_n'', w_1 and w_{i+1} which are four independent random variables:

$$\begin{aligned}
E_i &= \mathbb{E}\left[\frac{w_1 w_{i+1}}{(w_1 + W_n' + w_{i+1})(w_1 + W_n' + w_{i+1} + W_n'')}\right] \\
&= \int_0^\infty \mathbb{E}\left[\frac{w_1 w_{i+1}}{w_1 + W_n' + w_{i+1}} e^{-(w_1 + W_n' + w_{i+1})t}\right] \phi(t)^{n-i-1}\, dt \\
&= \int_0^\infty \int_t^\infty \mathbb{E}\left[w_1 w_{i+1} e^{-(w_1 + w_{i+1})u}\right] \phi(u)^{i-1}\phi(t)^{n-i-1}\, dt du \\
&= \int_0^\infty \int_t^\infty \phi'(u)^2 \phi(u)^{i-1}\phi(t)^{n-i-1}\, du dt\ .
\end{aligned}$$

Thus replacing the computation of E_i in the summation, we get:

$$\mathbb{E}[S_n] = 2\sum_{i=1}^{n-1} \int_0^\infty \int_t^\infty (n-i)\phi'(u)^2 \phi(u)^{i-1} \phi(t)^{n-i-1}\, du dt\ .$$

□

In the following theorem, we compute the moment of order 2 of the stationary search cost:

Theorem 2.2 *For a sequence ω of iid random weights, we have:*

$$\begin{aligned}\mathbb{E}[S_n^2] = \mathbb{E}[S_n] - 8\sum_{i=1}^{n-2}\sum_{j=1}^{n-i-1}(n-i-j)\times \\ \int_0^\infty\int_t^\infty\int_u^\infty \phi'(v)^2\phi'(u)\phi(v)^{i-1}\phi(u)^{j-1}\phi(t)^{n-i-j-1}\,dvdudt\ .\end{aligned} \tag{2.2}$$

Proof From [1] or [4], we have:

$$\mathbb{E}[S_n^2] = \mathbb{E}[\mathbb{E}[S_n^2\,|\,\omega]] = \mathbb{E}[S_n] + 4V\ ,$$

where V is the following quantity:

$$V = \sum_{1\leqslant i<j<k\leqslant n}\mathbb{E}\left[\frac{p_ip_jp_k}{p_i+\cdots+p_k}\left(\frac{1}{p_i+\cdots+p_j}+\frac{1}{p_j+\cdots+p_k}\right)\right]\ .$$

The expectation of S_n has been computing in the previous section (see theorem 2.1). Using that ω is a sequence of iid random weights, V can be rewritten as follows:

$$V = \sum_{i=1}^{n-2}\sum_{j=1}^{n-i-1}(n-i-j)\left[V_1(i,j)+V_2(i,j)\right]\ , \tag{2.3}$$

where:

$$V_1(i,j) = \mathbb{E}\left[\frac{p_1p_{i+1}p_{i+j+1}}{(p_1+\cdots+p_{i+j+1})(p_1+\cdots+p_{i+1})}\right]\ ,$$

and:

$$V_2(i,j) = \mathbb{E}\left[\frac{p_1p_{i+1}p_{i+j+1}}{(p_1+\cdots+p_{i+j+1})(p_{i+1}+\cdots+p_{i+j+1})}\right]\ .$$

Let us now compute these two expectations separately, using the same trick as in the proof of theorem 2.1. Let us start with the computing of the first expectation:

$$\begin{aligned}&V_1(i,j)\\ &= \mathbb{E}\left[\frac{p_1p_{i+1}p_{i+j+1}}{(p_1+\cdots+p_{i+j+1})(p_1+\cdots+p_{i+1})}\right]\\ &= \mathbb{E}\left[\frac{w_1w_{i+1}w_{i+j+1}}{(w_1+\cdots+w_{i+j+1})(w_1+\cdots+w_{i+1})W_n}\right]\\ &= \mathbb{E}\left[\frac{w_1w_{i+1}w_{i+j+1}}{(w_1+w'+w_{i+1}+w''+w_{i+j+1})(w_1+w'+w_{i+1})}\times\right.\\ &\qquad\qquad\left.\frac{1}{w_1+w'+w_{i+1}+w''+w_{i+j+1}+w'''}\right]\end{aligned}$$

where:

$$\left\{\begin{array}{lcl} w' &=& w_2+\cdots+w_i\\ w'' &=& w_{i+2}+\cdots+w_{i+j}\\ w''' &=& w_{i+j+2}+\cdots+w_n\ .\end{array}\right.$$

All these six random variables are independent. Thus we get:

$$
\begin{aligned}
& V_1(i,j) \\
&= \int_0^\infty I\!E\left[\frac{w_1 w_{i+1} w_{i+j+1}}{(w_1+w'+w_{i+1}+w''+w_{i+j+1})(w_1+w'+w_{i+1})} e^{-(w_1+w'+w_{i+1}+w''+w_{i+j+1})t}\right] \times \\
&\qquad\qquad \phi(t)^{n-i-j-1}\, dt \\
&= \int_0^\infty \int_t^\infty I\!E\left[\frac{w_1 w_{i+1} w_{i+j+1}}{w_1+w'+w_{i+1}} e^{-(w_1+w'+w_{i+1}+w''+w_{i+j+1})u}\right] \phi(t)^{n-i-j-1}\, dudt \\
&= -\int_0^\infty \int_t^\infty I\!E\left[\frac{w_1 w_{i+1}}{w_1+w'+w_{i+1}} e^{-(w_1+w'+w_{i+1})u}\right] \phi'(u)\phi(u)^{b-1}\phi(t)^{n-i-j-1}\, dudt \\
&= -\int_0^\infty \int_t^\infty \int_u^\infty I\!E\left[w_1 w_{i+1} e^{-(w_1+w'+w_{i+1})v}\right] \phi'(u)\phi(u)^{j-1}\phi(t)^{n-i-j-1}\, dvdudt \\
&= -\int_0^\infty \int_t^\infty \int_u^\infty \phi'(v)^2\phi'(u)\phi(v)^{i-1}\phi(u)^{j-1}\phi(t)^{n-i-j-1}\, dvdudt\ .
\end{aligned}
$$

Similar computations for the second expectation leads to the following expression:

$$
V_2(i,j) = -\int_0^\infty \int_t^\infty \int_u^\infty \phi'(v)^2\phi'(u)\phi(v)^{j-1}\phi(u)^{i-1}\phi(t)^{n-i-j-1}\, dvdudt\ .
$$

It is easy to see that $V_2(i,j) = V_1(j,i)$. Thus, we have:

$$
\sum_{i=1}^{n-2}\sum_{j=1}^{n-i-1}(n-i-j)V_2(j,i) = \sum_{i=1}^{n-2}\sum_{j=1}^{n-i-1}(n-i-j)V_1(i,j)\ .
$$

Hence we have the following expression for V:

$$
V = 2\sum_{i=1}^{n-2}\sum_{j=1}^{n-i-1}(n-i-j)V_1(i,j)\ .
$$

Thus replacing $V_1(i,j)$ by the computation we have done, we obtain the result stated in the theorem. □

Using the same techniques as below, one can obtain similar expressions for the two first moments when the weights are independent (but not necessary identical). For instance, one can prove the following result:

Theorem 2.3 *For a sequence ω of independent random weights, we have:*

$$
I\!E[S_n] = 2\sum_{1\leq i<j\leq n}\int_0^\infty \int_t^\infty \phi_i'(u)\phi_j'(u)\prod_{k=1}^{i-1}\phi_k(t)\prod_{k=i+1}^{j-1}\phi_k(u)\prod_{k=j+1}^{n}\phi_k(t)\, dudt\ . \tag{2.4}
$$

We have the following bound for the expected search cost:

Proposition 2.1 *For a sequence ω of independent random weights, we have:*

$$
I\!E[S_n] \leqslant 2\log n\ . \tag{2.5}
$$

This bound is not a surprise for the researcher in computer science since $\log n$ is the order of the height of a random binary search tree with n nodes.

Proof Allen and Munro proved the following inequality (theorem 3.2. in [1]):

$$
I\!E[S_n \,|\, \omega] \leqslant 2\log 2 H(\mathbf{p_n})\ , \tag{2.6}
$$

where:

$$
H(\mathbf{p_n}) = -\sum_{i=1}^{n} p_i \log_2 p_i\ ,
$$

is the Shannon entropy. It is well-known that:

$$
H(\mathbf{p_n}) \leqslant \log_2 n\ ,
$$

and that the maximum of the entropy is reached for the uniform distribution, that is when $p_i = 1/n$ for any $i \in \{1, \ldots, n\}$. Hence, we have the following inequality for the expected search cost:

$$\mathbb{E}[S_n] \leqslant 2 \log n \ .$$

□

Remark 2.1 This expected stationary search cost can be compared to the one when the data structure used is a list. In order to distinguish the two search cost, we introduce the following notations: let us denote by $S_n^{\mathcal{T}}$ (resp. $S_n^{\mathcal{L}}$) the stationary search cost when the data structure is a binary search tree updated with the move-to-root rule (resp. a list updated with the move-to-front rule). The case of a list with random weights was studied in [2]. From equation (2.1), since ϕ is a decreasing function of the time, one gets:

$$\mathbb{E}[S_n^{\mathcal{T}}] \leqslant 2 \sum_{1 \leq i < j \leq n} \int_0^\infty \left(\int_t^\infty \phi_i'(r) \phi_j'(r) \, dr \right) \prod_{\substack{k=1 \\ k \neq i,j}}^{n} \phi_k(t) \, dt \ ,$$

which equals to $\mathbb{E}[S_n^{\mathcal{L}}]$ (see corollary 1 in [2]). This result is not surprising since in the classical context the search cost of an element in a binary search tree is lower in expectation than in a list.

3 Examples

This section is devoted to study some examples. We will study three examples: deterministic, gamma and Poisson weights. For these two distribution we will be able to carry out all computations for the two first moments of S_n. Before, let us point out a useful remark when studying examples:

Remark 3.1 Let ω and ω' two sequences of independent strictly positive random variables. Assume that there exists a positive constant k such that for any $i \in \mathbb{N}^*$, $w_i' \overset{(d)}{=} k w_i$. Let n be an integer and consider the n first terms of these two sequences. Let $W_n = w_1 + \cdots + w_n$ and $W_n' = w_1' + \cdots + w_n'$. We construct two vectors of probabilities, $\mathbf{p_n}$ and $\mathbf{p_n'}$, as in the introduction. Then,

$$p_i' \overset{(d)}{=} \frac{k w_i}{\sum_{j=1}^{n} k w_j} = \frac{k w_i}{k W_n} = p_i \ .$$

Thus, according to theorem 3.1 in [1], the stationary search costs associated to the sequences ω and ω' are equal in distribution; or equivalently, all moments of the stationary search costs associated to these sequences are equal.

In the examples studied in this section, harmonic numbers will be used (see [14], p. 73-76). Hence let us recall its definition and some useful properties for computations:

Definition 3.1 For any $n \in \mathbb{N}^*$, the n-th harmonic number H_n is:

$$H_n = \sum_{k=1}^{n} \frac{1}{k} \ .$$

As n tends to infinity we have the following well-known approximation:

Lemma 3.1 *Let γ be the Euler's constant. As n tends to infinity,*

$$H_n = \log n + \gamma + \frac{1}{2n} - \frac{1}{12n^2} + \frac{1}{120n^4} - \epsilon \ ,$$

where:

$$0 < \epsilon < \frac{1}{252 n^6} \ ,$$

Let us now recall two formulas where harmonic numbers are involved in sum:

Lemma 3.2 *For any $n \in \mathbb{N}^*$,*

$$\sum_{k=1}^{n} H_k = (n+1) H_n - n \ .$$

and:

$$\sum_{k=1}^{n} \frac{1}{k} H_k = \frac{1}{2}(H_n^2 + H_n^{(2)}) \ ,$$

where:

$$H_n^{(2)} = \sum_{k=1}^{n} \frac{1}{k^2} \ .$$

We have the following limit for $H_n^{(2)}$:

Lemma 3.3

$$\lim_{n \to \infty} H_n^{(2)} = \frac{\pi^2}{6} \ .$$

One can easily prove the following lemma:

Lemma 3.4 *For any $n \in \mathbb{N}^*$ and $c \in \mathbb{Z}$,*

$$\sum_{k=1}^{n} \frac{k}{k+c} = n - c(H_{n+c} - H_c) \ .$$

If $c = 1$, it reduces to:

$$\sum_{k=1}^{n} \frac{k}{k+1} = (n+1) - H_{n+1} \ .$$

We have now all the tools to study our two examples:

Example 3.1 Let us consider a sequence of independent deterministic weights: for any $i \in \{1, \ldots, n\}$, $w_i = a_i$. The Laplace transform ϕ_i is:

$$\phi_i(s) = e^{-sa_i} \ .$$

Thus one gets:

$$\mathbb{E}[S_n] = 2 \sum_{1 \leq i < j \leq n} \frac{a_i a_j}{A_n(a_i + \cdots + a_j)} \ , \tag{3.1}$$

where $A_n = a_1 + \cdots + a_n$.

Let us now consider the case of iid weights. Without loss of generality, one can choose for any $i \in \{1, \ldots, n\}$, $a_i = 1$ using remark 3.1. Equation (3.1) reduces to:

$$\mathbb{E}[S_n] = 2 \left(1 + \frac{1}{n}\right) H_n - 4 \ . \tag{3.2}$$

Since weights are identical and deterministic, it is clear that this quantity is also the average number of comparison involved in a successful search if the keys are inserted into the tree on random order ([15], p. 427).

When the number n of items tends to infinity, we obtain the following asymptotic formula:

$$\mathbb{E}[S_n] \sim 2 \log n \ , \tag{3.3}$$

since $H_n \sim \log n$. Hence we find the formulas (equations (3.1), (3.2) and (3.3)) derived by Allen and Munro ([1], p. 529).

Let us now compute the moment of order 2 (in the iid case). One can obtain that:

$$\mathbb{E}[S_n^2] = 8 \left(1 + \frac{1}{n}\right) \left(H_n H_{n-1} - \frac{1}{2} H_n^2 + \frac{5}{4} H_n - \frac{1}{2} H_{n-1}^{(2)}\right) - 8 H_{n-1} + 20 - \frac{8}{n} \ .$$

Hence we obtain the asymptotic equivalent for the moment of order 2:

$$\mathbb{E}[S_n^2] \sim 4 \log^2 n \ .$$

Thus one can deduce that:

$$\frac{\mathrm{Var}[S_n]}{\mathbb{E}[S_n]^2} \xrightarrow[n \to \infty]{} 0 \ .$$

Using the Chebyshev inequality (see p. 233 in [11] for instance), we obtain that:

$$\frac{S_n}{I\!E[S_n]} \xrightarrow[n\to\infty]{Pr} 1 \ ,$$

which means that S_n is concentrated around its expectation as n becomes large. This property for S_n is surprising since the stationary search cost for a list updated with the move-to-front rule does not hold the same property (see corollary 2 and example 1 in [2]). One can believe that this is due to the kind of data structure used in this paper.

Example 3.2 Let us consider a sequence of independent random variables having the Gamma distribution with parameters a_i and λ. Without loss of generality, we can assume that $\lambda = 1$ using remark 3.1 (and since if X is a random variable having the Gamma distribution with parameters a and λ then λX has the Gamma distribution with parameters a and 1). The Laplace transform ϕ_i is:

$$\phi_i(s) = (1+s)^{-a_i} \ .$$

Thus one gets:

$$I\!E[S_n] = 2 \sum_{1\le i<j\le n} \frac{a_i a_j}{A_n(a_i + \cdots + a_j + 1)} \ , \tag{3.4}$$

where $A_n = a_1 + \cdots + a_n$.

Let us now consider the case of iid weights. Let $a_i = a$ for any $i \in \{1, \ldots, n\}$. Equation (3.4) reduces to:

$$\begin{aligned} I\!E[S_n] &= \frac{2a}{n} \sum_{i=1}^{n-1} \frac{n-i}{(i+1)a+1} \qquad (3.5) \\ &= 2\left(a + \frac{a+1}{n}\right) \sum_{i=1}^{n-1} \frac{1}{(i+1)a+1} - \frac{2(n-1)}{n} \ . \end{aligned}$$

Since $1/(ia+1) \sim 1/(ia)$ as i tends to infinity and since the series of general term $1/(ia)$ diverges, we have:

$$\sum_{i=1}^{n-1} \frac{1}{(i+1)a+1} \sim \frac{1}{a} \sum_{i=2}^{n} \frac{1}{i} \ , \tag{3.6}$$

as n tends to infinity. Thus we obtain the following approximation:

$$I\!E[S_n] \sim 2 \log n \ . \tag{3.7}$$

Notice that for this example, the distribution of $\boldsymbol{p_n}$ is well-known and is the symmetric Dirichlet distribution with parameter $(a, \ldots, a) \in \mathbb{R}_+^n$ (see [12] for instance). Let us recall that the density of this distribution is:

$$(x_1, \ldots, x_n) \mapsto \frac{\Gamma(an)}{\Gamma(a)^n} x_1^{a-1} \ldots x_n^{a-1} \mathbb{1}_{\Delta_n}(x_1, \ldots, x_n) \ ,$$

where Γ is the Gamma function and Δ_n is the simplex of order n:

$$\Delta_n = \left\{ (x_1, \ldots, x_n) \in \mathbb{R}_+^n \ ; \ \sum_{i=1}^{n} x_i = 1 \right\} \ .$$

For $a = 1$, the weights have the exponential distribution with parameter 1 and we obtain from equation (3.5):

$$I\!E[S_n] = 2\left(1 + \frac{2}{n}\right) H_{n+1} - 5 - \frac{4}{n} \ . \tag{3.8}$$

Let us now compute the moment of order 2 of the stationary search cost. One can get the following expression:

$$I\!E[S_n^2] = I\!E[S_n] + \frac{8a^2}{n} \sum_{i=1}^{n-2} \sum_{j=1}^{n-i-1} \frac{n-i-j}{(a(i+j+1)+1)(a(i+1)+1)} \ .$$

For $a = 1$ (exponential distribution), this expression reduces to:

$$\begin{aligned} I\!E[S_n^2] = 8H_{n+1}H_n - 4\left(1 + \frac{2}{n}\right) H_n^2 + \frac{16}{n} H_{n+2}H_n - 8\left(1 + \frac{1}{n}\right) H_n - 12H_{n+1} \\ -\frac{12}{n} H_{n+2} - 8\left(\frac{1}{2} + \frac{1}{n}\right) H_n^{(2)} + 14 + \frac{24}{n} \ . \end{aligned}$$

Hence we obtain the asymptotic equivalent for the moment of order 2:

$$\mathbb{E}[S_n^2] \sim 4\log^2 n \ .$$

Since we obtain the same asymptotic equivalents for the two first moments as in the case of deterministic weights, we can do the same remark about the convergence in distribution of S_n, that is:

$$\frac{S_n}{\mathbb{E}[S_n]} \xrightarrow[n\to\infty]{Pr} 1 \ .$$

Example 3.3 Let us consider a sequence of iid random variables having the Poisson distribution with parameter λ. The Laplace transform ϕ is:

$$\phi(s) = e^{\lambda(e^{-s}-1)} \ .$$

Thus one can compute the following asymptotic approximation:

$$\mathbb{E}[S_n] \sim 2\log n \ . \tag{3.9}$$

Hence we obtain the same approximation than in the deterministic and gamma case.

4 Conclusion and discussion

Throughout our theorems and the two first examples we studied, we can measure the performance of the binary search tree as a data structure used for retrieving quickly datas. Even we were not able to obtain a formula for the distribution of the stationary search cost (as in our previous work [2]), the elements contained in this paper can convince (one more time if needed?) of the advantage of binary search trees toward simple list. The interest of using binary search tree appears twice:

- The remark 2.1 succeeding to the theorem 2.1: the expected stationary search cost is always lower when using a binary search tree than using a simple list;
- The examples 3.1 and 3.2: for two families of distributions (deterministic and exponential), we obtain that the stationary search cost is concentrated around its expectation. This is not the case when using simple list (see examples in [2]): in this case the distribution of the stationary search cost is more spread.

Although comparisons between the two kinds of data structures can be done either for the expectation or for some examples, some questions leave unanswered and should merit to be underlying:

1. *Asymptotic formulas:* a question which arises naturally once formulas as equations (2.1) or (2.2) is obtained is about asymptotic equivalent and limit theorems. We give asymptotic equivalent for the three examples and it could be interesting to give an expression in the general case as it was done for the move-to-front rule (theorem 3 and corollary 2 of [2]). Notice that the technique used in [2] (approximation of Laplace integrals) cannot be applied in our case. In contrast with the case of a list (see section 4 of [2]), it seems that we obtain the same asymptotic equivalent for the expected search cost in the case of iid weights. A way to give a partial explanation is to do an integration by parts of equation (2.1). It follows that

   $$\mathbb{E}[S_n] = 2\left(H_{n-1} - \frac{n-1}{n}\right) - 2\sum_{i=1}^{n-1}\frac{n-i}{i}\int_0^\infty\int_t^\infty \phi(u)^i\phi''(u)\phi(t)^{n-i-1}\,dudt \ .$$

   The first part of the right-hand side is of order $2\log n$. Unfortunately we were not able to obtain an interesting inequality or an asymptotic equivalent for the second part of this equation.

2. *Bounds of the expectation of the entropy of random discrete distributions:* using a result of Allen and Munro [1], we obtained that the expected search cost is lower than the expectation of the entropy of the vector $\mathbf{p_n}$. Is it possible to obtain an inequality for $\mathbb{E}[H(\mathbf{p_n})]$ in order to obtain a sharper inequality for the expected search cost? Such inequality is especially interesting in the case of independent weights, since if the weights are iid, we have some result or indications about the behaviour of the expected search cost.

   To the best of our knowledge, few papers are dealing with this subject. Some work on this subject has been investigated by Shastri and Govil [18]. Kingman [13] proved that if the weights have the Gamma distribution

with parameter a and 1 and if there exists $\lambda > 0$ such that an tends to λ as n tends to infinity and a tends to 0, then we have:

$$\mathbb{E}[H(\mathbf{p_n})] \xrightarrow[n\to\infty]{} (\log 2)^{-1} \sum_{i=1}^{\infty} \frac{\lambda}{i(i+\lambda)} \ .$$

This limit reduces to:

$$(\log 2)^{-1} \left(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{\lambda}\right) \ ,$$

if λ is an integer. Darling [6] derived the two first moments of some functionals of $\mathbf{p_n}$ in the case of random weights exponentially distributed with parameter 1. For the entropy, he obtained that:

$$\mathbb{E}[H(\mathbf{p_n})] = -n(n+1) \int_0^1 (1-r)^{n-1} r \log_2(r)\, dr \ ;$$

and:

$$\begin{aligned}\mathbb{E}[H(\mathbf{p_n})^2] = &-n(n+1) \int_0^1 (1-r)^{n-1} r^2 \log_2(r)^2\, dr \\ &+n^2(n^2+1) \int_{\substack{r_1+r_2\leqslant 1 \\ r_1,r_2\geqslant 0}} (1-r_1-r_2)^{n-1} r_1 \log_2(r_1) r_2 \log_2(r_2)\, dr \ .\end{aligned}$$

Slud [20, 21] used these results to obtain a Central Limit Theorem for the entropy $H(\mathbf{p_n})$ of $\mathbf{p_n}$.

Another way to study it is to consider the entropy $H(\mathbf{p_n})$ conditionally to the sequence ω. For instance, from theorem 9.7.1 in [5] (p. 236), one can obtain the following bound:

$$\mathbb{E}[H(\mathbf{p_n})] \leqslant \frac{1}{2} \log(2\pi e) \left(\sum_{i=1}^{n} \mathbb{E}[i^2 p_i] - \mathbb{E}\left[\left(\sum_{i=1}^{n} i p_i \right)^2 \right] + \frac{1}{12} \right) \ .$$

One could express the expectations involved in this equation in terms of Laplace transform of the weights and obtain: for any $i \in \{1, \ldots, n\}$,

$$\mathbb{E}[p_i] = -\int_0^{\infty} \phi_i'(t) \prod_{\substack{k=1 \\ k\neq i}}^{n} \phi_k(t)\, dt \ ,$$

and:

$$\mathbb{E}[p_i^2] = \int_0^{\infty} \int_t^{\infty} \phi_i''(u) \prod_{\substack{k=1 \\ k\neq i}}^{n} \phi_k(u)\, du dt \ ,$$

and for any i, j such that $1 \leqslant i < j \leqslant n$,

$$\mathbb{E}[p_i p_j] = \int_0^{\infty} \int_t^{\infty} \phi_i'(u) \phi_j'(u) \prod_{\substack{k=1 \\ k\neq i}}^{n} \phi_k(u)\, du dt \ .$$

Unfortunately we do not obtain a nice formula from these expressions.

Acknowledgement We thank support from FONDAP-CONICYT in Applied Mathematics and Millenium Nucleus in Information and Randomness ICM P01-005. JB wish to thank CONICYT National Postgraduate Fellowship Program which support her PhD. CP wish to thank the program ECOS-Sud/CONICYT to support partially his post-doc at the CMM. Thanks to Béatrice Lachaud for her help in some computing. Thanks also to Thierry Huillet for pointing out references about entropy of random spacings.

References

[1] B. Allen and I. Munro. Self-organizing binary search trees. *J. Assoc. Comput. Mach.*, 25:526–535, 1978.

[2] J. Barrera and C. Paroissin. On the distribution of the stationary search cost for the move-to-front with random weights. *J. Appl. Prob.*. To appear in 2004.

[3] J. Bell and G. Gupta. An evaluation of self-adjusting binary search tree techniques. *Software - Practice and Experience*, 23(4):369-382, 1993.

[4] J. Bodell. *Cost of Searching - Probabilistic Analysis of the Self-Organizing Move-to-Front and Move-to-Root Sorting Rules*. PhD thesis, Mathematics Department, Royal Institute of Technology, Sweden, 1997.

[5] T.M. Cover and J.A. Thomas. *Elements of information theory*. Wiley, New-York, 1991.

[6] D.A. Darling. On a class of problems related to the random division of an interval. *Ann. Math. Statistics*, 24:239–253, 1953.

[7] R.P. Dobrow. The move-to-root rule for self-organizing trees with Markov dependent requests. *Stochastic Anal. Appl.*, 14(1):73–87, 1996.

[8] R.P. Dobrow and J.A. Fill. On the Markov chain for the move-to-root rule for binary search trees. *Ann. Appl. Probab.*, 5(1):1–19, 1995.

[9] R.P. Dobrow and J.A. Fill. Rates of convergence for the move-to-root Markov chain for binary search trees. *Ann. Appl. Probab.*, 5(1):20–36, 1995.

[10] P. Donnelly. The heaps process, libraries and size-biased permutations. *J. Appl. Prob.*, 28:321–335, 1991.

[11] W. Feller. *An introduction to the probability theory and its applications: volume I*. Wiley, New-York, 1968.

[12] L. Holst. Extreme value distributions for random coupon collector and birthday problems. *Extremes*, 4:129–145, 2001.

[13] J.F.C. Kingman. Random discrete distributions. *J. R. Statist. Soc.*, B37:1–22, 1975.

[14] D.E. Knuth. *The art of computer programming. Volume 1: fundamental algorithms*. Addison-Wesley Publishing Co., Reading, 1968.

[15] D.E. Knuth. *The art of computer programming. Volume 3: sorting and searching*. Addison-Wesley Publishing Co., Reading, 1973.

[16] V.G. Papanicolaou, G.E. Kokolakis, and S. Boneh. Asymptotics for the random coupon collector problem. *J. Comput. Appl. Math.*, 93:95–105, 1998.

[17] J. Pitman and M. Yor. Random discrete distributions derived from self-similar random sets. *Electron. J. Probab.*, 4(1):1–28, 1996.

[18] A. Shastri and R. Govil. Optimal discrete entropy. *Appl. Math. E-Notes*, 1:73–76, 2001.

[19] D.D. Sleator and R.E. Tarjan. Self-adjusting binary search trees. *J. Assoc. Comput. Mach.*, 32:652–686, 1985.

[20] E. Slud. Entropy and maximal spacings for random partitions. *Z. Wahrscheinlichkeitstheorie und Verw. Gebiete*, 41(4):341–352, 1978.

[21] E. Slud. Correction to: "Entropy and maximal spacings for random partitions". *Z. Wahrscheinlichkeitstheorie und Verw. Gebiete*, 60(1):139–141, 1982.