The List Update Problem: Improved Bounds for the Counter Scheme

Hadas Shachnai*
Dept. of Computer Science
The Technion – IIT
Haifa, 32000 ISRAEL

Micha Hofri†
Dept. of Computer Science
Rice University
Houston TX 77005

Abstract

We consider the problem of dynamic reorganization of a linear list, where requests for the elements are generated randomly with fixed, unknown probabilities. The objective is to obtain the smallest expected cost per access. It has been shown, that when no *a priori* information is given on the reference probabilities, the *Counter Scheme* (CS) provides an optimal reorganization rule, which applies to *all* possible distributions. In this paper we show that for a list of n elements, arbitrary request probabilities and $\alpha > 0$, the expected cost under CS achieves a ratio of at most $1+\alpha$ to the optimal (minimal) expected cost within $Qn\lg n/\alpha^2$ reorganization steps, for a Q we can compute.

1. Introduction

The list update problem was introduced by McCabe [11]: A fixed set of items is maintained as (an unsorted) linear list or as a serial file. Each request for an item requires a sequential search. The cost of accessing an item is determined by the length of this search. The list may be rearranged during a sequence of requests, so as to achieve a lower average access cost in subsequent requests. Assuming that each element may be accessed at any time with fixed probability, the optimal list arrangement is known: order the records by their access probabilities, in non-increasing order. In the present scenario these probabilities are not known, and our goal is to arrange the elements 'correctly,' in the above order.

A comprehensive survey of many *permutation rules* suggested for the list management and their probabilistic analyses appears in [5]. Later articles, which broadened the framework, are

*e-mail:hadas@cs.technion.ac.il
†e-mail: hofri@cs.rice.edu

[1, 7]. The first goes beyond the probabilistic approach, and the second introduces the topic of this paper. Recently, treatments of the most common rule, Move-to-Front, assuming the same access mechanism we use, and providing very detailed analyses, appeared in [3, 4].

The only yardstick used in the literature to compare different rules is the expected access cost; we abide by this choice. Rarely attention is given for higher moments, or tails of the distribution of the cost function (the references [3, 4] are unusual in this respect; they are analytical, rather than comparative, and illustrate well the difficulty of deriving informative distributional results here). This limitation maybe further defended as appropriate for the engineering point of view we adopt. It suggests there is little need for such information about the cost: the scenario envisioned for the use of such lists is that of numerous, frequent accesses. Since the cost is a finite random variable (bounded by the length of the list), rare deviations cannot be large, and are of small significance. It is the mean that is of interest, and in this situation the central limit theorem assures us that the mean gets quite close to the expected access cost, that we compute.

In this paper we focus on the *Counter Scheme* (CS), under which the list items are kept in non-increasing order of their reference counts, that are updated after every access to the list. Clearly, the counters are used to compute estimates for the unknown access probabilities. Ties are resolved by the rule that a referenced record is only moved forward as much as is needed to be ahead of all records with strictly smaller counters – it lands at the tail of the sublist of all records with counters equal to its own.

In [8] we showed that under the above conditions, the CS produces the least expected cost of access *at any stage*. Thus, the CS is time-optimal among all reorganization rules that use no *a priori* knowledge of the access probabilities. Compared with the cost of the optimal static order, the CS was shown in [7] to approach it within a factor of $1+\alpha$, for any $\alpha > 0$, in $Q_1(\alpha)n^2$ reorganization steps (for all possible reference probabilities), where n is the number of records in the list.

In the present work we improve this bound and show that, in fact, the expected access cost under the CS achieves a ratio of $1+\alpha$ to the minimum within $Q_2(\alpha)n\lg n$ reorganization steps (again, for a Q_2 we can compute, and which does not depend on the reference probabilities). This agrees with numerical studies, shown in part in [7]. Extensive calculations show that for several access distributions, the rate of increase of the bound is clearly super-linear in n, but still slightly below the claim. The basis of the improvement is a recently derived bound on tail-probability, relation (4) below, obtained in [2].

Unlike several other reorganization rules, the CS needs extra storage, to maintain the reference counters. Indeed, it appears that using the CS for an unlimited number of references needs counters of unbounded size. In an early discussion of the CS in [10, p. 398], Knuth suggests that the storage could be used otherwise, for additional information that allows nonsequential search. In practice, this access method is only used for short lists. For all human needs, the space overhead can be restricted to a single standard storage word (32 bits) per record (and in [7] we show excellent performance while using as few as 2 or 3 bits per record). In such circumstances nonsequential search is not likely to provide much advantage. In addition, there are applications where the search is not for a numerical relation (which can be usually converted to a nonsequential search), but for a non-arithmetic condition. Example [12]: a data communications routing program manages traffic in a node by selecting a path, for each message (or request for a circuit) it handles, from a given list; the paths are tested sequentially, until one with a suitable set of parameters (such as congestion levels at intermediate nodes) is found.

The paper is organized as follows: in section 2 we define more completely the model, and add detail to our assumptions. Section 3 contains the main results, and section 4 concludes with a discussion of open issues.

2. Preliminaries and Notation

We consider a linear list of n records, $L = \{R_1, \ldots, R_n\}$. Each record R_i is uniquely identified by a key $K_i, 1 \le i \le n$. Requests for records are drawn from a multinomial distribution specified by the reference probability vector (*rpv*): $\boldsymbol{p} = (p_1, \ldots, p_n)$. Thus, at each access, the request is for R_i in probability p_i, independently of the past, and in particular – of the state of the list[1]. This is known as the *independent reference model (irm)* [7, 12]. We assume, without loss of generality, a renumbering of the records, such that $p_1 \ge \cdots \ge p_n$.

Each reference requires a sequential search of the list. We define C, the cost of a single access, as the number of key-comparisons made till the specified record is reached. Under the *irm*, with a fixed *rpv*, the average access cost to the list is minimized when the records are in the optimal static order: R_i precedes R_j whenever $p_i > p_j$. This requires a complete knowledge of the *rpv*, or at least of the relative magnitude of the access probabilities. This knowledge is assumed unavailable.

The initial arrangement of L is assumed to be random, selected with equal probability out of all $n!$ possible permutations (this would be a reasonable assumption whenever the assembly of the list is not related to the mechanism that gives rise to the *rpv*). When the list is referenced, it is reorganized, with the aim of approaching the optimal ordering as the reference sequence grows longer.

In this work we derive results for the CS. Our performance measure is the expected access cost after the mth reference, $m \ge 0$, denoted by $C_m(CS \mid \boldsymbol{p})$.

Let σ_m denote the order of the list elements after the mth reference: $\sigma_m(i)$ is the position of R_i in the list. We write $\text{Prob}_{CS}(\sigma_m(j) < \sigma_m(i))$ for the probability that R_j precedes R_i after the mth reference, when the list is reorganized by the CS. Using this, the expected access cost under the CS after the mth reference is

$$C_m(CS \mid \boldsymbol{p}) = C(OPT \mid \boldsymbol{p}) + \sum_{1 \le i < j \le n} (p_i - p_j) \cdot \text{Prob}_{CS}(\sigma_m(j) < \sigma_m(i)), \tag{1}$$

[1] Absolute time plays no role here. We identify the arrivals of requests with ‘clock ticks.’

where

$$C(OPT \mid \boldsymbol{p}) = \sum_{i=1}^{n} i p_i = 1 + \sum_{1 \le i < j \le n} p_j, \tag{2}$$

is the expected access cost under the optimal static arrangement of the list. All the index pairs $(i < j)$ appear exactly once in this last summation.

By the strong law of large numbers [9, for example], for any $p_i > p_j$,

$$\lim_{m \to \infty} \text{Prob}_{CS}(\sigma_m(j) < \sigma_m(i)) = 0.$$

Hence

$$\lim_{m \to \infty} C_m(CS \mid \boldsymbol{p}) = C(OPT \mid \boldsymbol{p}).$$

What the law of large numbers does not provide is an estimate of the rate of convergence of $C_m(CS \mid \boldsymbol{p})$ to its limit. We measure the closeness of $C_m(CS \mid \boldsymbol{p})$ to its limit by their ratio. Our aim: to compute m, the number of references (= reorganization steps, in the counter scheme) so that the ratio $C_m(CS \mid \boldsymbol{p})/C(OPT \mid \boldsymbol{p})$ is below $1+\alpha$, for any specified $\alpha > 0$. Clearly, this m would normally depend both on α and the distribution $\boldsymbol{p}$; since $\boldsymbol{p}$ is unknown, there is interest in a global bound, that holds for all $\boldsymbol{p}$. This is what we propose to do.

The following lemma, which states the monotonicity property of $C_m(CS \mid \boldsymbol{p})$ is used in defining a stopping point for the reorganization process:

Lemma 1: [7] The cost function $C_m(CS \mid \boldsymbol{p})$ is monotone non-increasing in m for all $m \ge 1$.

Hence, given some $\alpha > 0$, once we find a number of steps, m^*, such that

$$C_{m^*}(CS|\boldsymbol{p}) \le (1+\alpha) C(OPT \mid \boldsymbol{p}), \tag{3}$$

then for all $m > m^*$, also $C_m(CS \mid \boldsymbol{p}) \le (1+\alpha) C(OPT \mid \boldsymbol{p})$.

The next lemma gives the desired stopping point when the access distribution is *assumed known* (possibly up to the mapping of probabilities to the keys). In the next section we use this result to derive a global, *distribution free* bound on the stopping point. The lemma is based on the bound shown in [2]:

$$\text{Prob}_{CS}(\sigma_m(j) < \sigma_m(i)) \le (1 - (\sqrt{p_i} - \sqrt{p_j})^2)^m. \tag{4}$$

Lemma 2: [2] For a given *rpv* $\boldsymbol{p}$ and any $0 < \alpha < 1$, the cost under CS achieves a ratio of $(1+\alpha)$ to $C(OPT \mid \boldsymbol{p})$ within m^* steps, where

$$m^* = \min_{m \ge 1} \{ m \mid \sum_{1 \le i < j \le n} (p_i - p_j)(1 - (\sqrt{p_i} - \sqrt{p_j})^2)^m \le \alpha (1 + \sum_{1 \le i < j \le n} p_j) \}. \tag{5}$$

Since the bound (4) is often quite loose, then for nearly any $\boldsymbol{p}$, lemma 2 prescribes a value for m^* which is much larger than is in fact required (as can be computed from the explicit formula for $C_m(CS \mid \boldsymbol{p})$ given in [7]). In various experiments ratios in the range 4 to 15+ have been observed, with the higher values corresponding to the larger α values, such as 0.2.

3. A Stopping Point for the CS

The following is our main result:

Theorem 1: For any $\alpha > 0$ and all *rpv* $\boldsymbol{p}$, $C_m(CS \,|\, \boldsymbol{p})$ approaches the optimal cost to within a factor $1+\alpha$ following a finite, precomputable number of reorganization steps, $m^*(n,\alpha)$, given by

$$m^*(n,\alpha) = \left\lceil \frac{n(1+\alpha)(1+\sqrt{1+\alpha})^2}{\alpha^2} \ln\left(\frac{n}{\alpha}\right) \right\rceil . \tag{6}$$

□

Proof: The proof is computational. Define the set of ordered pairs

$$S = \{(i,j) \,|\, 1 \le i < j \le n, \quad p_i - p_j > \alpha p_j\} . \tag{7}$$

Pairs which are not in S may be thought of as "too close to matter" (this is discussed in more detail in [7, 8], where we explain the statistical virtues of the CS).

By equation (5) it suffices to find the minimal $m \ge 1$, such that[2]

$$\sum_{1 \le i < j \le n} (p_i - p_j)(1 - (\sqrt{p_i} - \sqrt{p_j})^2)^m \le \alpha(1 + \sum_{(\dot{i},\dot{j}) \in S} p_j + \sum_{(\dot{i},\dot{j}) \notin S} p_j). \tag{8}$$

To obtain m^* from relation (8) we use the definition (7), dropping from both sides of (8) the contributions of pairs not in S (which satisfy the inequality for any $m \ge 0$), and are left with the (sufficient) requirement on m:

$$\sum_{(\dot{i},\dot{j}) \in S} (p_i - p_j)(1 - (\sqrt{p_i} - \sqrt{p_j})^2)^m \le \alpha(1 + \sum_{(\dot{i},\dot{j}) \in S} p_j). \tag{9}$$

We tighten the condition on m by dropping from the right-hand side the terms $\alpha \sum_{(\dot{i},\dot{j}) \in S} p_j$, to have the final requirement

$$\sum_{(\dot{i},\dot{j}) \in S} (p_i - p_j)(1 - (\sqrt{p_i} - \sqrt{p_j})^2)^m \le \alpha. \tag{10}$$

The following notation is useful:

$$V = \sum_{(\dot{i},\dot{j}) \in S} p_i, \quad N \equiv |S|, \quad \text{and} \quad A \equiv \sum_{(\dot{i},\dot{j}) \in S} (p_i - p_j). \tag{11}$$

We note that

$$\sum_{(\dot{i},\dot{j}) \in S} p_j = V - A. \tag{12}$$

[2] We use dots under indices that take part in the summation, when it may not be obvious.

Claim 1: For any $(i,j) \in S$

$$(\sqrt{p_i} - \sqrt{p_j})^2 > \left(1 - \frac{2}{1+\sqrt{1+\alpha}}\right)(p_i - p_j). \tag{13}$$

Proof: Let $p_i = q_{ij}^2 p_j$. For $(i,j) \in S$, definition (7) implies that $q_{ij}^2 > 1+\alpha$. Compute

$$\frac{(\sqrt{p_i} - \sqrt{p_j})^2}{p_i - p_j} = \frac{p_i + p_j - 2\sqrt{p_i p_j}}{p_i - p_j} = 1 - \frac{2}{q_{ij}+1},$$

which yields inequality (13). □

Let $d \equiv 1 - \frac{2}{1+\sqrt{1+\alpha}}$. Using relation (13), the left-hand side of relation (10) is bounded by $\sum_{(i,j)\in S}(p_i - p_j)(1 - d(p_i - p_j))^m$. We simplify the task of finding an upper bound for m, by "maximizing" this last expression. Specifically, while $A = \sum_{(i,j)\in S}(p_i - p_j)$, we consider *all* sequences $\{a_{ij} \geq 0\}$ such that $\sum_{(i,j)\in S} a_{ij} = A$, and look for one that with any given $m > 1$, maximizes the function

$$\sum_{(i,j)\in S} a_{ij}(1 - da_{ij})^m.$$

The maximum is obtained when $a_{ij} = \frac{A}{N}$ for all $(i,j) \in S$. Thus, it is sufficient to find the minimal m satisfying

$$\sum_{(i,j)\in S} \frac{A}{N}\left(1 - \frac{dA}{N}\right)^m < \alpha. \tag{14}$$

It remains to solve for the minimal m such that

$$\left(1 - \frac{dA}{N}\right)^m < \frac{\alpha}{A}. \tag{15}$$

We can write now an expression for m^*, but we need first to relate the values of A and N to the problem parameters, n and α.

Claim 2: For any $n \geq 2$ and A, N as defined in (11), $A/N \geq \alpha/[n(1+\alpha)]$.

Proof: Let

$$N_k \equiv |\{(i,j) \in S \;:\; j = k\}| \quad \text{and} \quad V_k \equiv \sum_{(i,k)\in S} p_i.$$

That is, for $2 \leq k \leq n$, N_k is the size of the subset of ordered pairs in S, in which the smaller probability is p_k. Clearly

$$N = \sum_{k=1}^{n} N_k \quad \text{and} \quad V = \sum_{k=1}^{n} V_k.$$

In addition, for any $(i,j) \in S$, the condition $p_i - p_j > \alpha p_j$ leads to

$$p_i - p_j > \alpha p_i/(1+\alpha). \tag{16}$$

Hence,

$$\frac{A}{N} = \frac{\sum_{k=1}^{n} \sum_{(i,k)\in S} (p_i - p_k)}{\sum_{k=1}^{n} N_k} > \min_{2\le k\le n} \frac{\sum_{(i,k)\in S} p_i \alpha/(1+\alpha)}{N_k} = \frac{\alpha}{1+\alpha} \min_{2\le k\le n} \frac{V_k}{N_k}.$$

The order $p_1 \ge \cdots \ge p_n$ implies that if $(i,k) \in S$, then for all $1 \le i' < i$ we also have $(i',k) \in S$. Therefore

$$\frac{V_k}{N_k} = \frac{p_1 + \cdots + p_{N_k}}{N_k},$$

i.e., the ratio V_k/N_k is the average of the N_k largest probabilities. This value is at least $1/n$. □

The last relation we need:

Claim 3: For any n and $\boldsymbol{p}$, $A < n$.

Proof: Clearly $A = \sum_{(i,j)\in S}(p_i - p_j) \le A' \equiv \sum_{1\le i<j\le n}(p_i - p_j) = \sum_{i=1}^{n}(n-2i+1)p_i$ which is at most $n-1$. □

Let $s \equiv \frac{\alpha}{1+\alpha}$. Using (15) and the above claims, we can bound m^* by solving the inequality

$$\left(1 - \frac{s\cdot d}{n}\right)^m < \frac{\alpha}{n}, \tag{17}$$

and $m = \left\lceil \frac{n}{sd} \ln\left(\frac{n}{\alpha}\right) \right\rceil$ provides the result stated in the theorem, since $sd = 1 + (1 - 2\sqrt{1+\alpha})/(1+\alpha)$. □

4. Discussion

The theorem above resolves a discrepancy that was evident in [7], between the bounds we could prove and the numerical evidence. Experiments with several probability mass functions, such as Zipf ($p_i = 1/iH_n$), linear ($p_i = 2(n-i+1)/[n(n+1)]$), and a few geometric distributions, showed that the required number of references to organize the list is close to the $n\log n$ pattern.

> In fact, the observed rate of increase was somewhat slower. It is possible that $n\sqrt{\log n}$ would be a better fit, but was hard to determine: evaluating $C_m(CS\,|\,\boldsymbol{p})$ once n exceeds 50 or so (and the interesting range for m then shoots up as well – super-linearly), is a complex numerical task. At stake is not only the computing time but also the achievable precision.

The results above are short of what we wanted in two ways. One is that the theorem claims that $m^*(n,\alpha) \propto \alpha^{-2}$, whereas the numerical experiments suggests that $m^*(n,\alpha) \propto \alpha^{-1}$, roughly. Changing our computational scheme to reproduce this behavior would be most interesting (or alternatively – discovering a counter-example, a $\boldsymbol{p}$ that achieves this bound!).

Second, the original motivation for looking for $m^*(n,\alpha)$ was to provide the user of a linear list with a guarantee: use the CS for this many steps, and rest assured that the access cost of the list is as close to the optimum as desired. But if this predicted number is in fact much larger than the needed number, by an order of magnitude, as seems to be the case (and sometimes more), the attractiveness of the scheme is diminished seriously. We are unsure yet how much of this difference is due to natural causes (a global bound is certain to provide a less-than-tight fit in most cases), and how much is due to our proof technique.

In view of this gap—regardless of its source—between the needed m^* and the global, guaranteed bound, we recommend that users who believe the *irm* is a good working hypothesis for their list, use the limited-counter-scheme, as described in [7], with the counter bound set to 3 or 4.

Can the gap be narrowed? We have commented about the looseness of relation (4) and its implication for (5). We have not seen a simple way to assess the effect of the device leading to (14). One suspected source, giving up the terms $\alpha\sum_{(\dot{i},\dot{j})\in S} p_j$ on the right-hand side of (9) does not seem to do much damage. To see the reason we first note that $\sum_{(\dot{i},\dot{j})\in S} p_j \le \sum_{1\le i<j\le n} p_j = C(OPT\,|\,\boldsymbol{p}) - 1$. While for *any* distribution likely to arise in practice, the last value is at least 1 (and could be a sizable fraction of n), it is easy to manufacture sequences of *rpv*s such that $C(OPT\,|\,\boldsymbol{p})$ gets arbitrarily close to 1, and the terms we deleted are as small as we desire. The simplest example is $p_2 = 1 - p_1 = n^{-r}$, and all other $p_j = 0$, with some integer r (to avoid trivialities with records that are never requested, we could instead assume these p_j are all equal to n^{-2r}, and reduce, say, p_2, by ever so little). Below we show that a small $C(OPT\,|\,\boldsymbol{p})$ can only arise from an *rpv* like this.

We say that not much of the gap is due to neglecting the above sum, since for such distributions, where we neglect very little, our bound is particularly poor. It is an easy claim, that these skewed distributions require relatively few references to organize properly – for the inequality (3) to hold (a claim that looks beyond the fact that for such concentrated *rpv*s the entire issue of reorganizing the list to improve its access cost is nearly meaningless).

To substantiate this claim we look at such a distribution, and show a bound on the α for which the inequality (3) is satisfied following a single reference.

Claim 4: Let

$$1+\frac{1}{n^{r+1}} \le C(OPT\,|\,\boldsymbol{p}) < 1+\frac{1}{n^r},$$

then relation (3) holds with $m = 1$ for all $\alpha > n^{-(r-1)}$.

Proof: By computation. From $C(OPT\,|\,\boldsymbol{p}) < 1+1/n^r$ we get $p_1 > 1-1/n^r$. Further, $C(OPT\,|\,\boldsymbol{p}) \ge 1+1/n^{r+1}$ implies there is a *maximal* value p_1 can have; the key observation is that when the sum of the access probabilities of the records R_2 through R_n is prescribed (at some $1-p_1$), they contribute to $C(OPT\,|\,\boldsymbol{p})$ the most when all the $p_j,\ j \in [2,n]$ are equal (to $(1-p_1)/(n-1)$). Then they contribute $(1-p_1)(n+2)/2$, and we find $p_1 \le 1-2/n^{r+2}$. Now

$$\begin{aligned} C_1(CS\,|\,\boldsymbol{p}) \;=\; & \Pr(R_1 \text{ is referenced}) E(C\,|\,R_1 \text{ is in position } 1) \\ & + \Pr(R_1 \text{ is not referenced}) E(C\,|\,R_1 \text{ is in a random position}). \end{aligned}$$

The various components are easy to bound:
$\Pr(R_1 \text{ is referenced}) \le 1-2/n^{r+2}$;
$E(C\,|\,R_1 \text{ is in position } 1)$ is at most $p_1+(1-p_1)(n+2)/2$, using the above consideration, so it is maximized when we use the above lower bound on p_1, and we obtain $1+1/(2n^{r-1})$;
$\Pr(R_1 \text{ is not referenced})$ is $1-p_1$, which is at most $1/n^r$;
$E(C\,|\,R_1 \text{ is in random position})$ gets a contribution from R_1 in an average position $(n+2)/2$, and the rest is again at most—we allow a slight overestimate—$(1-p_1)(n+2)/2$, for a total of $(n+2)/2$.

Collecting it all we find

$$C_1(CS\,|\,\boldsymbol{p}) \le \left(1-\frac{2}{n^{r+2}}\right)\left(1+\frac{1}{2n^{r-1}}\right)+\frac{1}{n^r}\frac{n+2}{2} = 1+\frac{1}{n^{r-1}}+\frac{1}{n^r}+ \text{ smaller terms.} \tag{18}$$

Hence the ratio $C_1(CS\,|\,\boldsymbol{p})/C(OPT\,|\,\boldsymbol{p})$ is bounded:

$$\frac{C_1(CS\,|\,\boldsymbol{p})}{C(OPT\,|\,\boldsymbol{p})} \le \frac{1+\frac{1}{n^{r-1}}+\frac{1}{n^r}}{1+\frac{1}{n^{r+1}}} \le 1+\frac{n^2+n+1}{n^{r+1}} \approx 1+\frac{1}{n^{r-1}}. \tag{19}$$

Hence the claim. □

We assumed throughout the paper that the initial state of the list is random, equally likely to be in any of the $n!$ states. While this appears to us natural in the context of computing average access cost, it may raise a question when we observe that we aim our treatment at providing a *guarantee* for the expected cost penalty. Evidently, the expected cost following m accesses depends on the initial state, but unless m is quite small, it depends very little: only the order of the tail of the list, of records that have not been referenced at all, is affected by the initial permutation. Otherwise the list is uniquely determined by the access process. Clearly when m approaches the values we are interested in, this tail is highly unlikely to contain anything beyond records with such small reference probabilities that we can afford to neglect its contribution to the expected cost. Hence the guaranteed claim is quite insensitive to the initial order.

Comment: In much of the literature about reorganizing lists the authors use asymptotic notation (such as big-O), when discussing functions of n or α. We do the same in [7]. We have not done so here, and suggest that when encountered in our previous work on this topic, the reader should understand it as a conventional abbreviation for an upper bound which is proportional to the given function. The reason is that using the sequential search lists is only reasonable for short to moderate lists (100 is quite long), and only moderate thresholds appear to us useful (for example, we think it is pointless to attempt for $\alpha < 0.01$). While we believe our results hold for lists of arbitrary length, and any threshold, we refrain from using the asymptotic notation because we want to emphasize the engineering aspects of this data structure and its algorithms.

ACKNOWLEDGMENT
The present version of the paper owes much to a close reading and "leading" questions of an anonymous referee (that led, i.a., to Claim 4).

References

[1] Bentley J. L., McGeoch C.C, "Amortized Analyses of Self-Organizing Sequential Search Heuristics", *Commun. ACM*, **28** (4) pp. 404–411, 1985.

[2] Cohen A., Rabinovich Y., Schuster A., Shachnai H., "Optimal Bounds on Tail Probabilities: A Study of an Approach", in *Randomized Parallel Computing*, edited by P. Pardalos and S. Rajasekaran, Kluwer Academic Publishers, 1998.

[3] Fill, J.A., "An Exact Formula for the Move-to-Front Rule for Self-Organizing Lists", *J. Theoret. Probab.* **9** pp. 113–160, 1996.

[4] Fill, J. A., "Limits and Rates of Convergence for the Distribution of Search Cost Under the Move-to-Front Rule", *Theor. Comput. Sci.* **164** pp. 185–206, 1996.

[5] Hester J.H., Hirschberg D.S., "Self-Organizing Linear Search", *ACM Comput. Surv.*, **17** (3) pp. 295–312, 1985.

[6] Hofri M., *Analysis of Algorithms*, Oxford University Press, New York 1995.

[7] Hofri M., Shachnai H., "Self-Organizing Lists and Independent References — a Statistical Synergy", *Jour. of Alg.*, **12** pp. 533–555, 1991.

[8] Hofri M., Shachnai H., "On the Optimality of Counter Scheme for Dynamic Linear Lists", *Inf. Process. Lett.*, **37** pp. 175–179, 1991.

[9] Karr A.F., *Probability*, Springer–Verlag, New York 1992.

[10] D.E. Knuth, *The Art of Computer Programming, Vol 3: Sorting and Searching.* Addison-Wesley, Reading MA 1973.

[11] McCabe J., "On Serial Files with Relocatable Records", *Operations Research*, **13** pp. 609–618, 1965.

[12] Topkis D. M., "Reordering Heuristics for Routing in Communication Network", *J. of Applied Prob.*, **23** pp. 130–143, 1986.