A TRANSPOSITION RULE ANALYSIS
BASED ON A PARTICLE PROCESS

DAVID GAMARNIK,* AND

PETAR MOMČILOVIĆ,** *IBM T. J. Watson Research Center*

Abstract

A linear list is a collection of items that can be accessed sequentially. The cost of a request is the number of items that need to be examined before the desired item is located, i.e. the distance of the requested item from the beginning of the list. The transposition rule is one of the algorithms designed to reduce the search cost by organizing the list. In particular, upon a request for a given item, the item is transposed with the preceding one. We develop a new approach for analyzing the algorithm, based on a coupling to a certain constrained asymmetric exclusion process. This allows us to establish an asymptotic optimality of the rule for two families of request distributions.

Keywords: Self-organizing list; average-case analysis; exclusion process

2000 Mathematics Subject Classification: Primary 68P05
Secondary 68P10; 60F99

1. Introduction

The linear list, a collection of items that can be accessed sequentially, is one of basic data structures known in computer science. A primary operation defined on the list is *search*. A requested item is found in the list by sequentially examining items from the beginning of the list. The *cost of search* is defined to be the distance of the requested item from the beginning of the list, i.e. the number of items that need to be examined in order to locate the desired item. Intuitively, one would like to place frequently requested items at the front of the list, so as to minimize the number of search steps. If the request sequence were known *a priori*, one could place items in an order that minimizes the search cost. Yet, often properties of the request sequence are either not known in advance or are time dependent. Hence, it is desirable to employ an algorithm that organizes the list based on past requests. The two best-known self-organizing algorithms are the *move-to-front rule* and the *transposition rule* [11, Section 6]. In addition to being simple, these rules are memory-free, i.e. they require no memory for their operation.

List-organizing algorithms have been analyzed over the past fifty years – see, for example, the review on self-organizing linear search in [8]. While the literature on the move-to-front rule (and the corresponding *least-recently-used caching algorithm*) is extensive (see, e.g. [3], [4], [5], [7], [9], [14], and references therein), results on the transposition rule are scarce. Early analysis of the transposition rule can be found in [13]. In that paper it was conjectured that the rule is optimal with respect to the expected value of the search cost. However, it was

shown in [1] that this conjecture is not true, in general. Except for the papers mentioned above, the probabilistic analysis of the transposition rule is either limited to the case of simplistic distributions [10], [15] or to numerical studies [2], [12]. The reader is referred to [14] for a combinatorial (amortized) analysis of the transposition rule.

In the present paper we develop a new approach for analyzing the transposition rule, based on a coupling to a constrained asymmetric exclusion process. This allows us to establish an asymptotic optimality of the rule for two families of request distributions. Specifically, we prove that the logarithm of the tail probability of the search cost is asymptotically optimal under the transposition rule when the request distribution is either power law or geometric.

The rest of the paper is organized as follows. The model description and main results can be found in the next section. In Section 3, we describe an associated asymmetric exclusion process and characterize its stationary behavior. In Section 4, we relate the exclusion process to the transposition rule for self-organizing lists. Section 5 contains the proofs of the results stated in Section 2. Our conclusions, and some open questions, are discussed in Section 6.

2. Model and results

We consider an infinite list of items $L = \{1, 2, \ldots, N, \ldots\} = \mathbb{N}$. At integer times $t = 0, 1, 2, \ldots$ a request arrives for an item from L. The item requested at time t is denoted by $R(t)$. The requests are independent and identically distributed, and π_i denotes the probability of item i being requested, such that $\sum_{i \geq 1} \pi_i = 1$. Without loss of generality, we assume that $\pi_i \geq \pi_{i+1}$ for all i. Let R be equal in distribution to $R(t)$, i.e. $\mathrm{P}[R = i] = \pi_i$.

The evolution of the list L is governed by the transposition rule. At time $t = 0$ the list is assumed to be ordered as $\{1, 2, \ldots, N, \ldots\}$. Upon every request, the requested object is moved forward (i.e. to the left) by one position in the list while the object in front of it is moved one position back (i.e. to the right). If the first item in L is requested, the list does not change. The basic idea is that frequently requested items are moved closer to the beginning of the list over time; on the other hand, items with low request probabilities end up at some distance from the beginning of the list.

At every time t, the list is represented as some permutation $\sigma : \mathbb{N} \to \mathbb{N}$. Let $X_i(t)$ be the position of the item i in the list at time t. Our focus is on the behavior of the position $C(t) := X_{R(t)}(t)$ of the requested item, i.e. the search cost, as $t \to \infty$. We note that if the permutation σ is fixed, then the distribution of $C(t)$ is determined completely by $\pi := \{\pi_i\}_{i=1}^{\infty}$. In this case, we use C^{σ} to denote the random position of the selected element R. Namely, C^{σ} is simply the (random) search cost required to locate the requested item in a list of *given* order σ. Thus, there exist two sources of randomness affecting the search cost: one due to the random arrangement σ of the items, and one due to the randomness of the requested item R.

Our first lemma is a simple observation stating that, for every permutation σ, the tail asymptotics of C^{σ} dominates the tail asymptotics of R.

Lemma 1. *For any distribution π, permutation σ, and for every $x \in \mathbb{N}$,*

$$\mathrm{P}[C^{\sigma} > x] \geq \mathrm{P}[R > x].$$

Proof. By the definition of the search cost,

$$\mathrm{P}[C^{\sigma} > x] = \sum_{\{j\,:\,\sigma(j) > x\}} \pi_j \geq \sum_{j > x} \pi_j = \mathrm{P}[R > x],$$

where the inequality holds by the monotonicity of the elements of π.

Thus, as far as the tail probability asymptotics is concerned, no list-ordering algorithm can achieve a better performance than the one under the optimal static arrangement. Note that arranging the items in (decreasing) order of π_i is feasible only if the distribution π is known in advance.

We say that R is distributed as a power law with parameter $\alpha > 1$ if $\pi_i = ci^{-\alpha}$ for all i, where $c^{-1} = \sum_{i \geq 1} i^{-\alpha}$ is the normalization constant. Random variable R is defined to be asymptotically geometric, with parameter $0 < \nu < 1$, when $i^{-1} \log \pi_i \to \log \nu$ as $i \to \infty$. The next result states that the transposition rule is asymptotically optimal with respect to the logarithm of the tail asymptotics for these two distribution families.

Theorem 1. *Let π be either power law with parameter $\alpha > 1$ or asymptotically geometric with parameter $0 < \nu < 1$. Then*

$$\limsup_{x \to \infty, \, t \to \infty} \frac{\log \mathrm{P}[C(t) > x]}{\log \mathrm{P}[R > x]} = 1.$$

Proof. See Section 5.

Often it is of interest to consider lists that contain only a finite number of items, i.e. such that π has a finite support. Although we will not make use of the following fact, we remark that, for every distribution π with finite support on $L_N = \{1, 2, \ldots, N\}$ (so that $\pi_i = 0$ for all $i > N$), the described system is an irreducible, reversible, aperiodic Markov chain, and the unique stationary solution is of the following product form:

$$\mathrm{P}[X_1 = i_1, X_2 = i_2, \ldots, X_N = i_N] = \frac{\prod_{j=1}^N \pi_j^{-i_j}}{\sum_{(k_1, \ldots, k_N) \in P_N} \prod_{j=1}^N \pi_j^{-k_j}},$$

where P_N denotes the set of all permutations of the list L_N. A natural way to introduce a power law and geometric distribution for the case of finite support is to take the distribution π conditioned on the event $\{i \leq N\}$. Denote by π_N the truncated distribution, and let the random variable R_N be defined by $\mathrm{P}[R_N > x] := \mathrm{P}[R > x \mid R \leq N]$. Note that the existence of a unique stationary distribution for every N allows us to consider the stationary search cost, denoted by C_N.

Theorem 2. *Let either* (i) *π_N be truncated power law with parameter $\alpha > 1$ and $x/N < \gamma$ for some $\gamma < 1$; or* (ii) *π_N be truncated asymptotically geometric with parameter $0 < \nu < 1$ and $x < N$. Then*

$$\lim_{\substack{x \to \infty \\ N \to \infty}} \frac{\log \mathrm{P}[C_N > x]}{\log \mathrm{P}[R_N > x]} = 1.$$

Proof. See Section 5.

3. Constrained asymmetric exclusion process

In this section, we consider a certain constrained asymmetric exclusion process in which we examine the deviation of the boundary particle from its minimal position. In particular, we consider an n-particle system on countably many slots on a half-line, enumerated from left to right as $1, 2, \ldots$. Each particle is associated with an independent Poisson process of unit intensity (the actual rate is not important). At the arrival times of their corresponding Poisson processes, particles move left or right with probability p or q, respectively. Multiple
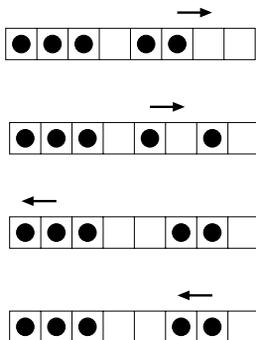
FIGURE 1: An example of evolution of the system with five particles. The arrows indicate intended movements of particles. Movements actually occur only in the upper two instances.

occupancies are not allowed, and a move actually occurs only if the target slot is empty (see Figure 1 for an illustration of this). A particle cannot move left if it is located in the first slot.

Assume that $p > q$ and define $\beta := q/p < 1$. Given that particles are enumerated from left to right using natural numbers, let Z_i be the position (slot number) of the ith particle. We first verify in a straightforward manner that the stationary distribution is of the following form:

$$P[Z_1 = i_1, Z_2 = i_2, \ldots, Z_n = i_n] = \eta_n^{-1} \beta^{\sum_{j=1}^n i_j} \tag{1}$$

for all $1 \leq i_1 < i_2 < \cdots < i_n$, where η_n is the normalization constant. To this end, notice that

$$\sum_{1 \leq i_1 < i_2 < \cdots < i_n} \beta^{\sum_{j=1}^n i_j} \leq \sum_{1 \leq j \leq n} \sum_{1 \leq i_j < \infty} \beta^{\sum_{j=1}^n i_j}$$

$$= \left(\frac{\beta}{1 - \beta} \right)^n < \infty,$$

implying that the normalization constant η_n is finite. It is easy to check that the underlying Markov chain is irreducible, reversible, and aperiodic, and that (1) satisfies the stationary equation. Hence, (1) indeed describes the stationary distribution.

We point out that the minimal possible value of $\sum_{i=1}^n Z_i$ is $\sum_{i=1}^n i = \frac{1}{2}n(n+1)$. Throughout the paper we interpret $\prod_{i=j}^k (\cdot) \equiv 1$ for $k < j$.

Lemma 2. *The normalization constant satisfies*

$$\eta_n = \beta^{n(n+1)/2} \prod_{i=1}^n \frac{1}{1 - \beta^i}.$$

Proof. For each integer $k \geq 0$, let $\eta_{n,k}$ denote the sum of $\beta^{\sum_{j=1}^n i_j}$ over the feasible choices of i_j such that $\max_{1 \leq j \leq n}\{i_j\} \leq n + k$. Then, clearly, $\eta_{n,0} \leq \eta_{n,1} \leq \cdots$ and $\eta_{n,k} \to \eta_n$ as $k \to \infty$. We claim that, for every n and k,

$$\eta_{n,k} = \beta^{n(n+1)/2} \prod_{i=1}^k \frac{1 - \beta^{n+i}}{1 - \beta^i}. \tag{2}$$

The expression for η_n follows immediately, by taking the limit as $k \to \infty$ in (2).

The proof of (2) is by induction. It is trivial to verify both that $\eta_{n,0} = \beta^{n(n+1)/2}$, $n \geq 1$, and $\eta_{1,k} = \sum_{i=1}^{k+1} \beta^i$, $k \geq 0$, conform to (2). Next, we assume that (2) holds for $\eta_{i,j}$ for all i and j such that either $i \leq n$ and $j < k$, or $i < n$ and $j \leq k$, and show that the statement is true for $\eta_{n,k}$. The quantity $\eta_{n,k}$ satisfies the following equality:

$$\eta_{n,k} = \eta_{n,k-1} + \beta^{n+k}\eta_{n-1,k}; \tag{3}$$

the first term corresponds to the case $\max_{1 \leq j \leq n}\{i_j\} < n + k$, and the second one to the case $\max_{1 \leq j \leq n}\{i_j\} = n+k$, i.e. the final ($n$th) particle is in the slot $n+k$. From (3) and the inductive assumption, we find that

$$\eta_{n,k} = \beta^{n(n+1)/2} \prod_{i=1}^{k-1} \frac{1 - \beta^{n+i}}{1 - \beta^i} + \beta^{(n-1)n/2+n+k} \prod_{i=1}^{k} \frac{1 - \beta^{n-1+i}}{1 - \beta^i}$$

$$= \left(1 + \beta^k \frac{1 - \beta^n}{1 - \beta^k}\right) \beta^{n(n+1)/2} \prod_{i=1}^{k-1} \frac{1 - \beta^{n+i}}{1 - \beta^i}$$

$$= \beta^{n(n+1)/2} \prod_{i=1}^{k} \frac{1 - \beta^{n+i}}{1 - \beta^i}.$$

This concludes the proof.

Next, we use (1) to examine the stationary deviation κ_n of the last particle from its minimal position, i.e. $\kappa_n := Z_n - n \geq 0$. Expressions (1) and (2), and Lemma 2, yield

$$P[\kappa_n = i] = \frac{\eta_{n-1,i}\,\beta^{n+i}}{\eta_n}$$

$$= \beta^i (1 - \beta^n) \prod_{j=1}^{n-1}(1 - \beta^{i+j})$$

$$< \beta^i \tag{4}$$

and, thus,

$$P[\kappa_n \geq i] < (1 - \beta)^{-1}\beta^i. \tag{5}$$

Interestingly, this implies that there exists a limiting behavior for the case when the number of particles n grows to infinity. Indeed, as $n \to \infty$, the random variable κ_n converges in distribution to a random variable κ with distribution given by

$$P[\kappa = i] = \beta^i \prod_{j=1}^{\infty}(1 - \beta^{i+j}).$$

From this, we conclude that κ is asymptotically geometric and stochastically monotone in the parameter β, and that

$$\lim_{i \to \infty} P[\kappa = i]\beta^{-i} = 1.$$

Finally, we note that for $\beta < \frac{1}{2}$ or, equivalently, for $2q < p$, the most probable value of κ is 0.

4. Coupling

The following lemma relates the stationary properties of list L, operating under the transposition rule and as $t \to \infty$, to characteristics of the particle system studied in Section 3. Let $\kappa_i(\beta)$ explicitly denote the dependency of the random variable κ_i on parameter β (see (4)), and set $\beta := \pi_{i+1}/\pi_i \le 1$.

Proposition 1. *For every* $x \ge i \ge 1$,

$$\limsup_{t\to\infty} \mathrm{P}\!\left[\bigvee_{j=1}^{i} X_j(t) > x\right] \le \mathrm{P}[\kappa_i(\beta) + i > x].$$

Remark 1. When the support of π is finite, there exists a unique stationary solution and, therefore, the left-hand side of the preceding inequality converges as $t \to \infty$.

4.1. Proof of Proposition 1

The proof is based on a coupling argument. We start by exploiting a Poisson embedding technique. (See [6] for an application of this technique in the context of the move-to-front rule.) The requests for item i form a Poisson process of intensity π_i, meaning that the limiting behavior (as $t \to \infty$) of the original discrete-time system is the same as that of the system with the Poisson request patterns.

Given a Poisson process (a set of arrival times) Λ with rate λ, let $\Lambda(p)$ denote its subset, $\Lambda(p) \subseteq \Lambda$, formed by including each element of Λ in $\Lambda(p)$ independently with probability p. Let $\{\Lambda_i\}$ be the set of request times for item $i \in \mathbb{N}$.

Next we construct a modified list $\hat{L}$ consisting of the same items as the original list L – parameters of the new system are denoted with the 'hat' symbol (caret). Specifically, $\hat{X}_j(t)$ denotes the position of element j at time t in the list $\hat{L}$. Each element $j \in \hat{L}$ is associated with an independent Poisson process $\hat{\Lambda}_j$ defined as

$$\hat{\Lambda}_j := \begin{cases} \Lambda_j(\pi_i/\pi_j), & 1 \le j \le i, \\ \Lambda_j \cup \Lambda_j^+, & j > i, \end{cases}$$

where Λ_j^+ is an independent Poisson process with rate $\pi_{i+1} - \pi_j$. Note that the processes $\hat{\Lambda}_j$ are constructed in such a way that they are Poisson, with rates π_i for $1 \le j \le i$ and π_{i+1} for $j > i$. In addition, observe that $\hat{\Lambda}_j \subseteq \Lambda_j$ for $1 \le j \le i$ and $\hat{\Lambda}_j \supseteq \Lambda_j$ for $j > i$. Furthermore, let the function $\varphi_j(t)$ be defined as follows:

$$\varphi_j(t) := \begin{cases} \mathbf{1}_{\{\exists\, k>i\,:\, \hat{X}_k(t)=\hat{X}_j(t)-1\}}, & j \le i, \\ \mathbf{1}_{\{\exists\, k\le i\,:\, \hat{X}_k(t)=\hat{X}_j(t)-1\}}, & j > i, \end{cases}$$

where $\mathbf{1}_{\{\cdot\}}$ denotes the indicator of the event $\{\cdot\}$. That is, the function $\varphi_j(t)$ indicates whether item j is preceded by an item k such that the rates of $\hat{\Lambda}_j$ and $\hat{\Lambda}_k$ differ.

The request process $\hat{\Lambda}$ of list $\hat{L}$ is a superposition of Poisson processes $\hat{\Lambda}_j$:

$$\hat{\Lambda}(t) := \bigcup_{\{j\,:\,\varphi_j(t)=1\}} \hat{\Lambda}_j(t). \tag{6}$$

Item j is requested from $\hat{L}$ at time $t = T$ if $T \in \hat{\Lambda}_j \cap \hat{\Lambda}$. In other words, item $j \le i$ is requested according to $\hat{\Lambda}_j$ only if it is preceded by an item in $\{i + 1, i + 2, \ldots\}$. On the other hand,
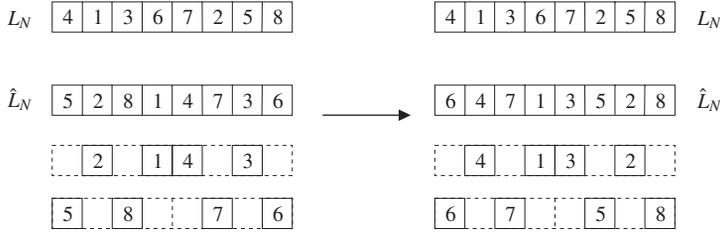
FIGURE 2: An example of reordering by the operator $\mathcal{R}_i$ (in the case $i = 4$ and $N = 8$). The initial states of the lists are shown on the left. The modified list $\hat{L}_N$ is divided into two sublists: $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$. Upon reordering items in each of the sublists according to L_N, the new ordering in $\hat{L}_N$ is shown on the right.

requests for item $j > i$ are placed according to $\hat{\Lambda}_j$ only if element j is preceded by an item in $\{1, \ldots, i\}$. Note that the set of Poisson processes included in $\hat{\Lambda}$ changes with the evolution of list $\hat{L}$. In addition, the number of elements in the union in (6) is always finite and bounded from above by $2i$.

The modified list $\hat{L}$ operates under the transposition rule with one modification. Namely, after the transposition rule rearranges items in either of the lists (L or $\hat{L}$), a reordering operator $\mathcal{R}_i$ is applied to $\hat{L}$. The operator works as follows. The list $\hat{L}$ is divided into two sublists: $\{1, 2, \ldots, i\}$ and $\{i + 1, i + 2, \ldots\}$. The operator $\mathcal{R}_i$ reorders each sublist so that the order of the elements within the sublists is the same as it is in the original list L. However, only items belonging to the same sublist are allowed to exchange positions in the list. An example of how $\mathcal{R}_i$ operates is shown in Figure 2.

Next, assume that both lists are in the same permutation at time $t = 0$. We then appeal to the following lemma.

Lemma 3. *For every $t \in \mathbb{R}_+$ and $1 \leq j \leq i$,*

$$X_j(t) \leq \hat{X}_j(t). \tag{7}$$

Proof. The lists change only at times of request to the corresponding system; denote these times by $0 < T_1 < T_2 < \cdots < T_n < \cdots$. Since there are no changes in item order between the times $\{T_n\}$, it is sufficient to prove that (7) holds for $t = T_n+$, $n \geq 1$. To this end, suppose that (7) holds for $t = T_{n-1}+$ and consider the two lists at time $t = T_n+$. The following three cases must be examined.

Case 1. At time $t = T_n$, item $j \leq i$ is requested from L. By the construction of $\hat{\Lambda}$, this event implies that item j is requested in the modified list $\hat{L}$ only with some probability depending on the state of $\hat{L}$.

If item j is not requested from $\hat{L}$, then the set of positions occupied by items $\{1, \ldots, i\}$ in $\hat{L}$ remains the same. On the other hand, the set of positions occupied by items $\{1, \ldots, i\}$ in L either does not change (when an item in $\{1, \ldots, i\}$ precedes item j or j is the first item in the list) or is pairwise smaller (when an item in $\{i + 1, \ldots, N\}$ precedes item j). This, in conjunction with the fact that the relative order of items $\{1, \ldots, i\}$ in L and $\hat{L}$ at $t = T_n+$ is the same (due to the action of $\mathcal{R}_i$), implies that (7) holds for $t = T_n+$.

On the other hand, if item j is requested in both lists, then the only case that needs to be examined in detail is the one in which j is preceded by item $l \leq i$ in L; note that, from (6), it

follows that j is preceded by an item $v > i$ in $\hat{L}$. Since $X_l(T_n-) \leq \hat{X}_l(T_n-)$ and the order of the items $\{1, \ldots, i\}$ is the same in both lists, we have

$$X_j(T_n-) + 1 \leq \hat{X}_j(T_n-). \tag{8}$$

The fact that items j and l are transposed in L but not in $\hat{L}$ means that the order of j and l is different in the two lists before $\mathcal{R}_i$ is applied. Thus, $\mathcal{R}_i$ exchanges the positions of items j and l in $\hat{L}$. This means that

$$\hat{X}_j(T_n+) = \hat{X}_l(T_n-) \geq X_l(T_n-) = X_j(T_n+)$$

and

$$\hat{X}_l(T_n+) = \hat{X}_j(T_n-) - 1 \geq X_j(T_n-) = X_l(T_n+),$$

where the inequality in the first equation follows from the inductive assumption, and that in the second is due to (8). Thus, we conclude that (7) holds for $t = T_n+$.

Case 2. At time $t = T_n$, item $j > i$ is requested from $\hat{L}$. The argument is very similar to that in Case 1. In this case, j is preceded by some $v \leq i$ in $\hat{L}$ (see (6)).

If item j is not requested from L or is preceded (in L) by an item $v > i$, then the positions occupied by items $\{1, \ldots, i\}$ in L do not change. However, in $\hat{L}$, item j must be preceded by $v \leq i$ and, thus, v is moved one position back by the transposition rule. Alternatively, if in L, item j is requested and preceded by $l \leq i$, then either $v = l$ and $X_l(T_n-) = \hat{X}_v(T_n-)$, or $v \neq l$ and

$$X_l(T_n-) \leq \hat{X}_l(T_n-) + 1,$$

$$X_v(T_n-) \leq \hat{X}_v(T_n-) + 1.$$

In either case, after the items are transposed we have

$$X_l(T_n+) \leq \hat{X}_l(T_n+)$$

and

$$X_v(T_n+) \leq \hat{X}_v(T_n+).$$

Case 3. At time $t = T_n$, item $j > i$ is requested from L but not from $\hat{L}$. This implies that j is preceded in $\hat{L}$ by an item $v > i$ (see (6)). If j is preceded in L by an item $l > i$, then the positions occupied by items $\{1, \ldots, i\}$ do not change in either list. Therefore, we need only consider the case in which j is preceded by $l \leq i$. However, in that case, we necessarily have

$$\hat{X}_l(T_n-) \geq X_l(T_n-) + 1,$$

which implies that

$$X_l(T_n+) = X_l(T_n-) + 1 \leq \hat{X}_l(T_n-) = \hat{X}_l(T_n+).$$

The results of these three cases establish (7), proving Lemma 3.

Next, let the variables $Z_j(t)$, $1 \leq j \leq i$, be defined by

$$(Z_1(t), \ldots, Z_i(t)) := \mathscr{S}(\hat{X}_1(t), \ldots, \hat{X}_i(t)),$$

where $\mathscr{S}$ is the sorting (in increasing order) operator. Note that the definition of $Z_i(t)$ and (7) imply that

$$Z_i(t) \geq \bigvee_{j=1}^{i} X_j(t). \tag{9}$$

Observe that the evolution of $\{Z_j(t)\}$ is probabilistically the same as in the constrained particle system described in Section 3 (recall that there $Z_j(t)$ denotes the position of the jth particle at time t), with $p = \pi_i/(\pi_i + \pi_{i+1})$ and $q = \pi_{i+1}/(\pi_i + \pi_{i+1})$. Indeed, $Z_j(t)$ increases by 1 at Poisson rate π_{i+1} only if $Z_{j+1}(t) \neq Z_j(t) + 1$, and it decreases by 1 at rate π_i only if $Z_{j-1} \neq Z_j(t) - 1$.

Taking maximums on both sides of (9), and applying the operator $\mathrm{P}[\cdot > x]$, leads to

$$\mathrm{P}\left[\bigvee_{j=1}^{i} X_j(t) > x\right] \leq \mathrm{P}[Z_i(t) > x],$$

from which we obtain

$$\limsup_{t \to \infty} \mathrm{P}\left[\bigvee_{j=1}^{i} X_j(t) > x\right] \leq \mathrm{P}[Z_i > x]$$

$$= \mathrm{P}[\kappa_i(\beta) + i > x],$$

where the last equality follows from the definition of $\kappa_i(\beta)$ in Section 3 and the fact that $\beta = \pi_{i+1}/\pi_i$.

5. Proofs

Proposition 1 is the primary tool in establishing our results on the performance of the transposition rule. The following lemma is a simple consequence of Proposition 1.

Lemma 4. *For any $y \geq 1$ and distribution of requests π,*

$$\limsup_{t \to \infty} \mathrm{P}[C(t) > x] \leq \mathrm{P}[\kappa_y(\pi_{y+1}/\pi_y) > x - y] + \mathrm{P}[R > y, R + \kappa_R(\pi_{R+1}/\pi_R) > x].$$

Proof. Conditioning on the requested item, and using the monotonicity of the max operator, results in

$$\limsup_{t \to \infty} \mathrm{P}[C(t) > x] = \limsup_{t \to \infty} \left(\sum_{i \geq 1} \pi_i \, \mathrm{P}[X_i(t) > x]\right)$$

$$\leq \sum_{i=1}^{y} \pi_i \limsup_{t \to \infty} \mathrm{P}\left[\bigvee_{j=1}^{y} X_j(t) > x\right] + \sum_{i \geq y+1} \pi_i \limsup_{t \to \infty} \mathrm{P}\left[\bigvee_{j=1}^{i} X_j(t) > x\right]$$

$$\leq \mathrm{P}[\kappa_y(\pi_{y+1}/\pi_y) > x - y] + \mathrm{P}[R > y, R + \kappa_R(\pi_{R+1}/\pi_R) > x],$$

where the last inequality follows from Proposition 1.

At this point we present the proofs of Theorems 1 and 2.

5.1. Proof of Theorem 1

The lower bound is an immediate consequence of Lemma 1 and holds for any distribution of requests π. Hence, we only consider the upper bound.

We first examine the case in which π is asymptotically geometric with parameter ν. Fix an arbitrary, small $\varepsilon > 0$ such that $\nu + \varepsilon < 1$. By the assumption, there exists i_ε such that $\nu - \varepsilon < \pi_{i+1}/\pi_i < \nu + \varepsilon$ for all $i \geq i_\varepsilon$. For any $s^{-1} > \nu + \varepsilon$, setting $y = i_\varepsilon$ in Lemma 4 yields

$$\limsup_{t \to \infty} P[C(t) > x] \leq P[\kappa_{i_\varepsilon}(\nu + \varepsilon) > x - i_\varepsilon] + P[R + \kappa_R(\nu + \varepsilon) > x]$$

$$\leq P[s^{\kappa_{i_\varepsilon}(\nu+\varepsilon)} > s^{x-i_\varepsilon}] + P[s^{R+\kappa_R(\nu+\varepsilon)} > s^x]$$

$$\leq s^{-x} s^{i_\varepsilon} \, E \, s^{\kappa_{i_\varepsilon}(\nu+\varepsilon)} + s^{-x} \, E \, s^{R+\kappa_R(\nu+\varepsilon)}, \tag{10}$$

where the last step is due to Markov's inequality. From $s^{-1} > \nu + \varepsilon$ and (5), it follows that $E \, s^{R+\kappa_R(\nu+\varepsilon)} < \infty$. This bound, (5), and (10) result in

$$x^{-1} \limsup_{t \to \infty} \log P[C(t) > x] \leq -\log s + x^{-1} \log(s^{i_\varepsilon} \, E \, s^{\kappa_{i_\varepsilon}(\nu+\varepsilon)} + E \, s^{R+\kappa_R(\nu+\varepsilon)})$$

$$\to -\log s \quad \text{as } x \to \infty. \tag{11}$$

On the other hand, note that

$$P[R > x] \geq \sum_{i > x} \pi_{i_\varepsilon} (\nu - \varepsilon)^{i-i_\varepsilon}$$

$$\geq \pi_{i_\varepsilon}(1 - \nu + \varepsilon)^{-1}(\nu - \varepsilon)^{x+1-i_\varepsilon},$$

implying that

$$\liminf_{x \to \infty} x^{-1} \log P[R > x] \geq \log(\nu - \varepsilon). \tag{12}$$

By combining (11) and (12), we obtain

$$\limsup_{x \to \infty, \, t \to \infty} \frac{\log P[C(t) > x]}{\log P[R > x]} \leq \frac{\log s^{-1}}{\log(\nu - \varepsilon)};$$

letting first $s^{-1} \downarrow \nu + \varepsilon$ and then $\varepsilon \downarrow 0$ yields the result.

Next, we consider the case in which π is power law with parameter $\alpha > 1$. Lemma 4 and (5) yield

$$\limsup_{t \to \infty} P[C(t) > x] \leq P[\kappa_y(\pi_{y+1}/\pi_y) > x - y] + P[R > y]$$

$$\leq \left(1 - \frac{\pi_{y+1}}{\pi_y}\right)^{-1} \left(\frac{\pi_{y+1}}{\pi_y}\right)^{x-y} + P[R > y].$$

Letting $y = \lceil \varepsilon x / \log x \rceil$ for a sufficiently small $\varepsilon > 0$ (where $\lceil \cdot \rceil$ denotes the smallest integer greater than or equal to its argument) results in an estimate on the two terms in the preceding expression: as $x \to \infty$,

$$\left(1 - \frac{\pi_{y+1}}{\pi_y}\right)^{-1} \left(\frac{\pi_{y+1}}{\pi_y}\right)^{x-y} = \alpha^{-1} e^\alpha y x^{-\alpha/\varepsilon}(1 + o(1))$$

and

$$P[R > y] = \sum_{i > y} c i^{-\alpha} = O(y^{-\alpha+1}).$$

Therefore, as $x \to \infty$,

$$\limsup_{t \to \infty} P[C(t) > x] \le P[R > y](1 + o(1)).$$

The preceding equation, together with the fact that π is a power law, yields the statement of the theorem.

5.2. Proof of Theorem 2

This proof is very similar to that of Theorem 1 and, thus, we omit the details. Since

$$\frac{\log P[C_N > x]}{\log P[R_N > x]} = \frac{\log P[C_N > x]}{\log P[R > x]} \frac{\log P[R > x]}{\log P[R_N > x]}$$

and the upper bound on $P[C_N > x]$ is the same as the one on $\limsup P[C(t) > x]$, we need only verify that the final fraction in the preceding equality tends to 1 as $x \to \infty$ and $N \to \infty$. However, this fact follows easily from the assumptions of the theorem.

6. Concluding remarks

In this paper, we presented an analysis of the transposition rule based on a coupling to a constrained exclusion process. As an outcome, we established an asymptotic optimality of the transposition rule in linear lists. Specifically, when the probability distribution of the requests is power law or geometric, we showed that, under the transposition rule, the logarithm of the tail probability of the search cost is asymptotically optimal.

While the steady-state distribution of the search cost is a primary quantity of interest, in practice, rates of convergence play an important role in assessing the applicability of self-organizing algorithms. The proposed coupling may offer new directions for understanding these rates under the transposition rule. The same question, for the related move-to-front algorithm, was investigated in [5]. As was remarked there, the transposition rule is expected to have slower rates of convergence than the move-to-front rule.

References

[1] ANDERSON, E., NASH, P. AND WEBER, R. (1982). A counterexample to a conjecture in optimal list ordering. *J. Appl. Prob.* **19,** 730–732.

[2] BACHRACH, R. AND EL-YANIV, R. (1997). Online list accessing algorithms and their applications: Recent empirical evidence. In *Proc. Eighth Annual Symp. Discrete Algorithms* (New Orleans, LA), ACM, New York, pp. 53–62.

[3] BARRERA, J. AND PAROISSIN, C. (2004). On the distribution of the search cost for the move-to-front rule with random weights. *J. Appl. Prob.* **41,** 250–262.

[4] FILL, J. (1996). An exact formula for the move-to-front rule for self-organizing lists. *J. Theoret. Prob.* **9,** 113–159.

[5] FILL, J. (1996). Limits and rates of convergence for the distribution of search cost under the move-to-front rule. *Theoret. Comput. Sci.* **164,** 185–206.

[6] FILL, J. AND HOLST, L. (1996). On the distribution of search cost for the move-to-front rule. *Random Structures Algorithms* **8,** 179–186.

[7] FLAJOLET, P., GARDY, D. AND THIMONIER, L. (1992). Birthday paradox, coupon collector, caching algorithms and self-organizing search. *Discrete Appl. Math.* **39,** 207–229.

[8] HESTER, J. AND HIRCHBERG, D. (1985). Self-organizing linear search. *Comput. Surveys* **17,** 295–311.

[9] JELENKOVIĆ, P. (1999). Asymptotic approximation of the move-to-front search cost distribution and least-recently-used caching fault probabilities. *Ann. Appl. Prob.* **9,** 430–464.

[10] KAN, Y. AND ROSS, S. (1980). Optimal list order under partial memory constraints. *J. Appl. Prob.* **17,** 1004–1015.

[11] KNUTH, D. (1998). *The Art of Computer Programming*, Vol. 3, 2nd edn. Addison-Wesley, Reading, MA.

[12] LAM, K., LEUNG, M. Y. AND SIU, M. K. (1984). Self-organizing files with dependent accesses. *J. Appl. Prob.* **21,** 343–359.

[13] RIVEST, R. (1976). On self-organizing sequential search heuristics. *Commun. ACM* **19,** 63–67.

[14] SLEATOR, D. AND TARJAN, R. (1985). Amortized efficiency of list update and paging rules. *Commun. ACM* **28,** 202–208.

[15] TENENBAUM, A. AND NEMES, R. (1982). Two spectra of self-organizing sequential search algorithms. *SIAM J. Comput.* **11,** 557–566.