J. Appl. Prob. **19**, 730–732 (1982)
Printed in Israel
0021–9002/82/030730–03 $00.55

A COUNTEREXAMPLE TO A CONJECTURE ON OPTIMAL LIST ORDERING

E. J. ANDERSON,*
P. NASH** AND
R. R. WEBER,*** *University of Cambridge*

Abstract

A number of items are arranged in a line. At each unit of time one of the items is requested, the ith being requested with probability P_i. We consider rules which reorder the items between successive requests in a fashion which depends only on the position in which the most recently requested item was found. It has been conjectured that the rule which always moves the requested item one closer to the front of the line minimizes the average position of the requested item. An example with six items shows that the conjecture is false.

OPTIMAL LIST ORDER; MEMORY CONSTRAINTS; TRANSPOSITION RULE

1. A conjecture on optimal list ordering

In modeling the storage of computer files, Rivest (1976) considered the problem of finding an optimal rule for self-ordering a line of items, $e_1, \cdots, e_n$. At each unit of time a request is made (independently of previous requests) to retrieve one of the items: the ith being requested with probability P_i, $P_i \geqq 0$, $\Sigma P_i = 1$. The retrieval cost is equal to the position of the requested item in the line. Items may be reordered without cost between requests, but it is not possible to remember the frequencies with which the different items have been requested in the past. Therefore any reordering which is made between two successive requests, must be made using a *no-memory rule* which reorders the items in a fashion depending only on the position in which the most recently requested item was found.

Rivest considered the *transposition rule* which transposes the requested item with the item which is one position closer to the front of the line. He showed that for all probability vectors $P = (P_1, \cdots, P_n)$ the transposition rule has a smaller

---

Received 20 July 1981.
* Postal address: Clare College, Cambridge CB2 1TL, U.K.
** Postal address: Churchill College, Cambridge CB3 0DS, U.K.
*** Postal address: Queens' College, Cambridge CB3 9ET, U.K.

average cost than the rule which replaces the requested item at the front of the line. He provided examples to support his conjecture that the average cost using the transposition rule is less than the average cost that can be obtained using any other no-memory rule. The work of Kan and Ross (1980) also supports the conjecture. They have shown that when $P_1 = p$, $P_2 = \cdots = P_n = (1-p)/(n-1)$ the transposition rule is optimal amongst the class of rules which just move the requested item some number of places closer to the front of the line.

2. A counterexample

We begin with an intuitive argument showing why the conjecture is false. Suppose n is very large, and $P_1 > P_2$ are the only non-zero request probabilities. Consider a rule which is identical to the transposition rule, except that following a request for an item which is found in the second position no reordering occurs, and following a request for an item which is found in the third position the front three items are permuted $(123) \to (231)$. Suppose that items e_1 and e_2 are initially in positions far from position 1 (which happens with high probability if n is large and the items are randomly ordered at the start). Then there is a high probability that e_1 will reach position 2 before e_2 reaches position 3. If this happens then e_1 and e_2 will eventually reach positions 1 and 2 respectively, where they will remain thereafter. Otherwise (when with small probability e_2 reaches position 2 before e_1 reaches position 3) they will eventually reach positions 1 and 2 in the opposite order. By taking n sufficiently large we can ensure that with a probability as near 1 as we like this rule will place e_1 and e_2 (for almost all time) in positions 1 and 2 respectively — their best possible positions. In the long run, the transposition rule places e_1 and e_2 in positions 1 and 2 respectively for only a proportion P_1 of the time. Thus the transposition rule has a greater expected average cost than that of the alternative rule, and this goes part of the way to showing the conjecture is false. Although intuitive, this does not fully demonstrate that the conjecture is false since the alternative rule does not generate an irreducible Markov chain, and the transposition rule is better than the alternative rule if e_1 and e_2 start off in positions 2 and 1 respectively. These undesirable features are eliminated in the counterexample that follows.

The counterexample is based on the previous remarks and concerns six items with request probabilities

$$P = (0.85, 0.146, 0.001, 0.001, 0.001, 0.001).$$

To describe a no-memory rule we must specify a set of six permutations $\pi_1, \cdots, \pi_6$, in which π_i is the permutation to be applied to the items in the various positions following a request of the item in the ith position. Consider the rule R with the following set of permutations:

$$\pi_1 = \begin{pmatrix} 123456 \\ 123456 \end{pmatrix} \qquad \pi_3 = \begin{pmatrix} 123456 \\ 231456 \end{pmatrix} \qquad \pi_5 = \begin{pmatrix} 123456 \\ 123546 \end{pmatrix}$$

$$\pi_2 = \begin{pmatrix} 123456 \\ 123456 \end{pmatrix} \qquad \pi_4 = \begin{pmatrix} 123456 \\ 124356 \end{pmatrix} \qquad \pi_6 = \begin{pmatrix} 123456 \\ 341265 \end{pmatrix}.$$

Let R^* be the transposition rule. Its average cost can be calculated from the stationary probabilities of the resulting Markov chain. These are easy to calculate since the Markov chain is time reversible. The average cost is 1.263704 (to six places of accuracy). Using rule R the average cost is 1.216094 — about 4 percent less. This cost can be found by computing the average cost over the first N units of time and continuing the computation until convergence is observed. The rule R generates an irreducible Markov chain and is better than R^* no matter how the items are ordered initially.

Further calculations have shown that R is only better than R^* when the probabilities are close to those used in the example. It seems reasonable that R^* should be optimal if the request probabilities are approximately equal. It is still an open question whether R^* is optimal in the class of no-memory rules which just move the requested item some number of places closer to the front of the line.

References

Kan, Y. C. and Ross, S. M. (1980) Optimal list order under partial memory constraints. *J. Appl. Prob.* **17**, 1004–1015.

Rivest, R. (1976) On self-organizing sequential search heuristics. *Comm. ACM* **19**, 63–67.