



## A Linear Search Problem

Eugene Wong

*SIAM Review*, Vol. 6, No. 2 (Apr., 1964), 168-174.

Stable URL:

<http://links.jstor.org/sici?sici=0036-1445%28196404%296%3A2%3C168%3AALSP%3E2.0.CO%3B2-V>

*SIAM Review* is currently published by Society for Industrial and Applied Mathematics.

---

Your use of the JSTOR archive indicates your acceptance of JSTOR's Terms and Conditions of Use, available at <http://www.jstor.org/about/terms.html>. JSTOR's Terms and Conditions of Use provides, in part, that unless you have obtained prior permission, you may not download an entire issue of a journal or multiple copies of articles, and you may use content in the JSTOR archive only for your personal, non-commercial use.

Please contact the publisher regarding any further use of this work. Publisher contact information may be obtained at <http://www.jstor.org/journals/siam.html>.

Each copy of any part of a JSTOR transmission must contain the same copyright notice that appears on the screen or printed page of such transmission.

---

JSTOR is an independent not-for-profit organization dedicated to creating and preserving a digital archive of scholarly journals. For more information regarding JSTOR, please contact [support@jstor.org](mailto:support@jstor.org).

## A LINEAR SEARCH PROBLEM\*

EUGENE WONG†

### SUMMARY

A class of one-dimensional search problems is considered. In general, the formulation results in a functional-minimization equation of the dynamic programming type. In a special case the optimal solution for both the objective and policy have been found.

### INTRODUCTION

WE CONSIDER a storage unit consisting of  $N$  cells, with information stored in tabular form. That is, the record  $r(i)$  stored in cell  $i$  is the form of argument-function  $[x_i, f(x_i)]$ , the file being arranged in ascending order of the argument  $x_i$ . An example of such an arrangement is a dictionary.

Give a particular argument  $x$ , we find  $f(x)$  by searching for the cell containing  $[x, f(x)]$ . The search proceeds by comparing  $x$  against the arguments in a sequence of cells  $i_1, i_2$ , etc. This sequence is to be chosen so as to minimize the average number of comparisons required for locating the correct cell. Problems of this kind occur in addressing computer storage [1].

### II. EQUATION OF OPTIMIZATION

We begin with the following assumptions.

(1) In a comparison of  $x$  against  $x_i$ , only three possible outcomes exist, namely,

$$x > x_i, \quad x < x_i, \quad x = x_i.$$

(2) Let  $\xi$  be an integer-valued random variable denoting the location of  $x$ . We assume that the *a priori* probabilities  $p_k = \text{Prob} [\xi = k]$  are given, with

$$(1) \quad \sum_{k=1}^N p_k = 1$$

(3) Let  $S$  be the set of integers 1 through  $N$ , and let  $\sigma$  be a non-empty subset of  $S$ . We assume that the *a posteriori* probability distribution of  $\xi$  is unchanged except for renormalization; i.e.,

$$(2) \quad \begin{aligned} \text{Prob} [\xi = k \mid \xi \in \sigma] &= \frac{p_k}{P(\sigma)}, k \in \sigma \\ &= 0, k \notin \sigma \end{aligned}$$

where  $P(\sigma) = \sum_{i \in \sigma} p_i$ .

\* This work was supported by the Air Force Office of Scientific Research of the Office of Aerospace Research; the Department of the Army, Army Research Office; and the Department of the Navy, Office of Naval Research, under Grant No. AF-AFOSR-139-63; and National Science Foundation Grant No. 21292. Received by the editors August 9, 1963 and in revised form December 15, 1963.

† Dept. of Electrical Engineering and Electronics Research Laboratory, University of California, Berkeley, California.



There is not a unique policy  $n^*(N)$  which yields the minimum. In fact, the multiplicity of solutions can be quite large. The complete set of solutions is

$$(8) \quad \begin{aligned} n^*(2^{k+1} + 2m) &= 2^k + j, & j &= 0, 1, \dots, 2m + 1, & m < 2^{k-1} \\ & & j &= 2m - 2^k + 1, \dots, 2^k, & m \geq 1^{k-1} \\ n^*(2^{k+1} + 2m - 1) &= 2^k + 2j, & j &= 0, 1, \dots, m, & m \leq 2^{k-1} \\ & & j &= m - 2^{k-1}, \dots, 2^{k-1}, & m > 2^{k-1} \end{aligned}$$

For example, consider  $N = 2^4 + 9 = 25$

$$n^*(N) = 10, 12, 14, 16.$$

The policy solution is interesting and somewhat surprising. Intuitively, one would expect that the optimal solution  $n^*(N)$  should be such as to divide the remaining  $N - 1$  cells into nearly equal subsets, i.e.,  $N - n^* \cong n^* - 1$ . Thus, the large multiplicity of solution is not expected. Furthermore, in some cases the midpoint is in fact not a solution. For example, for  $N = 25$ , the point  $n = 13$  divides the remaining 24 cells equally, but is not among the solutions.

#### IV. PROOF OF OPTIMALITY

In this section we shall prove that the solutions of (6) are indeed given by (7) and (8). The proof proceeds in three stages. First, it is shown that the right-hand side of (6) is minimized by a specific choice of policy  $n^*(N)$ . Next,  $T(N)$  will be derived. Finally, the multiplicity of the policy solution is found.

A. If we let  $f(N) = NT(N)$ , (6) is simplified and can be rewritten as

$$(9) \quad f(N) = N + \min_{1 \leq n \leq N} \{f(n-1) + f(N-n)\}.$$

We begin by proving the following theorem:

**THEOREM 1:** *Under the conditions  $f(0) = f(1) = 0$ , the minimization in (9) is achieved with  $n = n^*(N)$ , where for all positive integers  $m$ ,*

$$(10) \quad n^*(4m-2) = n^*(4m-1) = n^*(4m) = n^*(4m+1) = 2m.$$

*Proof:* It seen that Theorem 1 is equivalent to the following set of equations with  $m$  ranging over all positive integers:

$$(11a) \quad f(4m-2) = 4m-2 + f(2m-2) + f(2m-1)$$

$$(11b) \quad f(4m-1) = 4m-1 + f(2m-1) + f(2m-1)$$

$$(11c) \quad f(4m) = 4m + f(2m-1) + f(2m)$$

$$(11d) \quad f(4m+1) = 4m+1 + f(2m-1) + f(2m+1).$$

We proceed by induction. First, by enumerating all possibilities, we find that (11) is true for  $m = 1$ . Next, we assume (11) to be true for  $m = 1, \dots, K$  and prove the following lemma:

**LEMMA:** *Equation (11) being valid for  $m = 1, 2, \dots, K$ , implies*

$$(12) \quad f(n + 1) - f(n) \geq f(n - 1) - f(n - 2), \quad n = 2, \dots, 4K$$

$$(13) \quad f(n + 1) > f(n), \quad n = 1, 2, \dots, 4K$$

$$(14) \quad f(2n) - f(2n - 1) > f(2n + 1) - f(2n), \quad n = 1, 2, \dots, 2K.$$

*Proof:* If (11) is true for  $m = 1, \dots, K$ , then

$$(15) \quad \begin{aligned} & [f(4m + 1) - f(4m)] - [f(4m - 1) - f(4m - 2)] \\ &= [f(2m + 1) + f(2m - 2)] - [f(2m) + f(2m - 1)], \quad m = 1, 2, \dots, K. \end{aligned}$$

Now under the same assumption, [compare (11c) and (9)],

$$(16) \quad f(2m) + f(2m - 1) = \min_{1 \leq n \leq 4m} \{f(n - 1) + f(4m - n)\}, \quad 1 \leq m \leq K.$$

Therefore, it follows that

$$(17) \quad f(2m + 1) + f(2m - 2) \geq f(2m) + f(2m - 1), \quad 1 \leq m \leq K,$$

and

$$(18) \quad f(4m + 1) - f(4m) \geq f(4m - 1) - f(4m - 2), \quad 1 \leq m \leq K.$$

Similarly, we find that

$$(19) \quad f(4m - 1) - f(4m - 2) = f(4m - 3) - f(4m - 4)$$

$$(20) \quad f(4m) - f(4m - 1) \geq f(4m - 2) - f(4m - 3)$$

$$(21) \quad f(4m - 2) - f(4m - 3) = f(4m - 4) - f(4m - 5), \quad m \leq K.$$

Relationships (18) through (21) imply (12). Further, since  $f(2) - f(1) > 0$  and  $f(3) - f(2) > 0$ , inequality (12) implies that  $f(n + 1) - f(n) > 0$ , which is inequality (13).

To prove (14), we note that if (11) is valid for  $m = 1, \dots, K$ , then

$$f(4m - 2) - f(4m - 3) = 1 + f(2m - 2) - f(2m - 3)$$

$$f(4m - 1) - f(4m - 2) = 1 + f(2m - 1) - f(2m - 2)$$

$$f(4m) - f(4m - 1) = 1 + f(2m) - f(2m - 1)$$

$$f(4m + 1) - f(4m) = 1 + f(2m + 1) - f(2m).$$

It is clear that  $f(2m) - f(2m - 1) > f(2m + 1) - f(2m)$  for  $m = 1, 2, \dots, K$  implies

$$f(4m - 2) - f(4m - 3) > f(4m - 1) - f(4m - 2),$$

and

$$f(4m) - f(4m - 1) > f(4m + 1) - f(4m), \quad m = 1, 2, \dots, K.$$

Therefore, inequality (14) follows from  $f(2) - f(1) > f(3) - f(2)$ . This latter is easily verified.

Proceeding with the main part of the proof for Theorem 1, we write  $f(4K + 2)$  as

$$(22) \quad \begin{aligned} f(4K + 2) &= 4K + 2 + \min_{2 \leq n \leq 4K+1} [f(n - 1) + f(4K + 2 - n)] \\ &= 2K + 2 + \min \left\{ \min_{1 \leq n \leq K} [f(2n - 1) + f(4k + 2 - 2n)], \right. \\ &\quad \left. \min_{1 \leq n \leq K} [f(2n) + f(4K + 1 - 2n)] \right\}. \end{aligned}$$

Using (12), we find that

$$f(n + 1) + f(n - 2) \geq f(n) + f(n - 1), \quad n = 2, \dots, 4K.$$

Thus,

$$f(2K + 2) + f(2K - 1) \leq f(2K + 3) + f(2K - 2) \leq \dots f(4K) + f(1),$$

and

$$f(2K + 2) + f(2K - 1) = \min_{1 \leq n \leq K} [f(2n - 1) + f(4K + 2 - 2n)].$$

Similarly, we find that

$$f(2K) + f(2K + 1) = \min_{1 \leq n \leq K} [f(2n) + f(4K + 1 - 2n)].$$

Therefore, (22) is reduced to

$$\begin{aligned} f(4K + 2) &= 4K + 2 + \min \{ [f(2K + 2) + f(2K - 1)], [f(2K) + f(2K + 1)] \} \\ &= 4K + 2 + f(2K) + f(2K + 1), \end{aligned}$$

where the last step follows again from (12). We note that we have extended (11a) to  $m = K + 1$ , and (12) to  $n = 4K + 1$ .

Using (12) and (13) and arguments similarly to those in reducing (22), we can write  $f(4K + 3)$  as

$$(23) \quad f(4K + 3) = 4K + 3 + \min \{ 2f(2K + 1), f(2K) + f(2K + 2) \}$$

It follows from (14) that

$$f(2K + 2) - f(2K + 1) \geq f(2K + 3) - f(2K + 2),$$

and it follows from (12) that

$$f(2K + 3) - f(2K + 2) > f(2K + 1) - f(2K).$$

Therefore,

$$f(2K + 2) + f(2K) > 2f(2K + 1),$$

and from (23)

$$(24) \quad f(4K + 3) = 4K + 3 + 2f(2K + 1).$$

Following a procedure nearly identical to the above, we can show that

$$(25) \quad f(4K + 4) = 4K + 4 + f(2K + 1) + f(2K + 2),$$

and

$$(26) \quad f(4K + 5) = 4K + 5 + f(2K + 1) + f(2K + 3).$$

By induction, Theorem 1 follows.

B. The functional form of  $f(N)$  is given by the following theorem:

THEOREM 2. Equation (9) is satisfied if and only if

$$(27a) \quad \begin{aligned} f(2^{k+1} + 2m - 1) &= 2^{k+1}(k - \frac{1}{2}) + 2mk + 3m + 1, \quad k = 0, 1, \dots, \\ m &= 0, 1, \dots, 2^k, \end{aligned}$$

$$(27b) \quad \begin{aligned} f(2^{k+1} + 2m) &= 2^{k+1}(k - \frac{1}{2}) + (2m + 1)k + 3m + 3, \quad k = 0, 1, \dots \\ m &= 0, 1, \dots, 2^{k-1}. \end{aligned}$$

*Proof:* The “only if” part follows simply from the fact that no two functions can both be the minimum without being equal. To prove (27), we again use induction. That is, we verify (27) for  $k = 0$  and assume it to be valid for  $k = 0, 1, \dots, K - 1$ . If it follows thereby that (27) is valid for  $k = K$ , then (27) must be true for all  $k$ . The detailed proof involves substitution of (27) in (11) and elementary manipulation, and will be omitted here.

C. Theorem 1 is strengthened by the following result:

$$(28) \quad f[n^*(N) - 1] + f[N - n^*(N)] = \min_{1 \leq n \leq n} \{f(n) + f(N - n)\}$$

if and only if

$$(29a) \quad \begin{aligned} n^*(2^{k+1} + 2m) &= 2^k + j, \quad j = 0, 1, \dots, 2m + 1, \quad 0 \leq m < 2^{k-1} \\ j &= 2m - 2^k + 1, \dots, 2^k, \quad 2^{k-1} \leq m < 2^k - 1, \end{aligned}$$

$$(29b) \quad \begin{aligned} n^*(2^{k+1} + 2m - 1) &= 2^k + 2j, \quad j = 0, 1, \dots, m, \quad 0 \leq m \leq 2^{k-1} \\ j &= m - 2^{k-1}, \dots, 2^{2k-1}, \quad 2^{k-1} \leq m \leq 2^k. \end{aligned}$$

*Proof:* The “if” part is proved by substituting (29) and (27) in (28) and verifying. In the process it is also shown that for  $2^{k+1} \leq N \leq 2^{k+2} - 1$ , the only solution in the range  $2^k \leq n^* \leq 2^{k+1}$  are those given by (29). Thus, it remains only to show that no value of  $n^*$  greater than  $2^{k+1}$  or less than  $2^k$  is a solution.

Consider  $N = 2^{k+1} + 2m, 0 \leq m < 2^{k-1}$ . Since we know that  $n^* = 2^k$  is a solution, we need only to show that (similar results follow for  $n^* > 2^{k+1}$  by symmetry)

$$(30) \quad \begin{aligned} f(2^k - 1) + f(2^k + 2m) &< f(2^k - 2) + f(2^k + 2m + 1) \\ &\leq f(2^k - 3) + f(2^k + 2m + 2) \leq \dots \end{aligned}$$

The first of the inequalities in (30) is easily verified using (27). The remaining inequalities follow from (12). For  $2^{k-1} \leq m \leq 2^k$ , we use  $n^* = 2^{k+1}$ , and from

(27) and (12) show that

$$(31) \quad f(2^{k+1} - 1) + f(2m) < f(2^{k+1}) + f(2m - 1) \leq f(2^{k+1} + 1) \\ + f(2m - 2) \leq \dots$$

Inequalities (30) and (31) suffice to prove (29a).

For  $N = 2^{k+1} + 2m - 1$ , the proof of (29b) follows nearly identical lines and will not be reproduced here.

#### REFERENCES

1. W. W. PETERSON, *Addressing for random-access storage*, IBM Jour. of Res. and Dev., 1 (1957), pp. 130-146.
2. R. BELLMAN, *Dynamic Programming*, Princeton University Press, Princeton, N. J., 1957.