Playing Twenty Questions with a Procrastinator*

Andris Ambainis[†] Stephen A. Bloch[‡] David L. Schweizer[§]

October 26, 1998

Abstract

We study the classic binary search problem, with a delay between query and answer. For all constant delays, we give upper and lower bounds, matching up to an additive constant.

1 Introduction

Recently one of us (S. Bloch) found himself talking with a colleague about how to match homework assignments to the intellectual level of an unknown group of students. An obvious algorithm appeared to be binary search: give a homework assignment, then adjust the next assignment up or down in difficulty depending on the students' performance on this assignment.

However, the course in question was a seminar that met once a week. One homework problem was assigned each week, due in class the following week. As a result, there was no chance to grade the students' performance on assignment m until after giving assignment $m+1$. In other words, each query must be chosen before the previous one is answered. The problem obviously generalizes to any fixed delay $d \in \mathbf{N}$. Trivially, this form of search is at most a factor of $d+1$ slower than the usual form ($d=0$): simply ask the same query $d+1$ times in a row, ignoring all but the first answer. (This is roughly equivalent to giving homework only every $d+1$ weeks, which the students would no doubt prefer.) The question remained: is this slowdown of $d+1$ optimal? We show that the optimal slowdown is in fact $\log_{\varphi_d} 2$, where φ_d is the positive real root of $x^{d+1} = x^d + 1$. But first, a formal definition of our problem.

Definition 1. *The discrete-real[1] search problem with delay d and size n*

Instance: *a function $f : \boldsymbol{Z} \to \boldsymbol{R}$, a domain $[a,b]$ (where $b = a+n-1$) on which f is presumed nondecreasing, and a real y_0 such that $f(a) < y_0 < f(b)$.*

Output: *either an integer $c \in [a,b]$ such that $f(c) = y_0$, or a pair of adjacent integers $c, c+1 \in [a,b]$ such that $f(c) < y_0 < f(c+1)$.*

Allowed operations:[2] *the i-th query consists of an integer x_i, and is answered by one of $f(x_i) < y_0$, $f(x_i) = y_0$, or $f(x_i) > y_0$. For all i, the answer to query i is available before the algorithm chooses the value of x_{i+d+1}.*

As our title suggests, this problem is related to Ulam's famous question [Ula76] about finding a number between one and one million by asking questions of an opponent who may lie once or twice. Various specializations and generalizations of this problem have been studied (*e.g.,* [Pel87, Pel89, Guz90, DGW92, Aig96]). None of this work has examined delayed answers.

The problem also appears to be related to the problem of finding the maximum of a unimodal function. The first solutions [Kie53, OW64, AW66a], in which Fibonacci searching was developed, were extended to parallel searching [AW66a, KM68] and to searching with a delay [BW69]. More recently, this work has been extended to different searching rules [GR93] and generalized to k-modal function optimization [MR96]. Although the two problems are related, they are not (see below) isomorphic.

Definition 2. *The discrete-real unimodal optimization problem with delay d and size n*

Instance: *a function $f : \boldsymbol{Z} \to \boldsymbol{R}$, and a domain $[a,b]$ where $b = a+n-1$ on which f is presumed unimodal (*i.e.* for some $c \in [a,b]$, f is increasing on $[a,c)$ and decreasing on $(c,b]$).*

Output: *an integer $c \in [a,b]$ such that for all integers $d \in [a,b]$, $f(c) \geq f(d)$.*

[†]Computer Science Division, University of California, Berkeley, CA 94720. Email: ambainis@cs.berkeley.edu.

[‡]Department of Math & Computer Science, Adelphi University, Garden City, NY 11530. Email: sbloch@adelphi.edu.

[§]Barclays Global Investors, 45 Fremont Street, San Francisco, CA 94105. Email: dls@aya.yale.edu.

[1]We can also consider the same problem for functions defined on reals instead of integers. It is handled similarly in the full version of our paper.

[2]Lance Fortnow points out that if arbitrary subset queries are allowed, the problem becomes uninteresting: the algorithm can simply query each bit of the answer, and since these queries are nonadaptive anyway, the delay makes little difference and the optimal algorithm takes exactly $\lceil \log_2(n) \rceil$ queries.

Allowed operations: *the i-th query consists of an integer x_i, and is answered with the value of $f(x_i)$. For all i, the answer to query i is available before the algorithm chooses the value of x_{i+d+1}.*

The unimodal optimization problem with delays $d = 1, 2, 3$ was studied by Beamer and Wilde[BW69] and Li[Li84]. Using a complicated case-by-case analysis, they show that $\log_{\psi_d} n + o(\log_{\psi_d} n)$ queries are necessary and sufficient, where $\psi_0 = (\sqrt{5}-1)/2 = 1.618...$, $\psi_1 = \sqrt{2} = 1.4145...$, $\psi_2 \approx 1.325$, and $\psi_3 \approx 1.2786$.

The monotone search problem reduces easily to the unimodal search problem (simply minimize the function $f(x)^2$). However, the converse does not hold, and it is easy to see that unimodal optimization requires more queries than search in the delay-0 case. Our results together with [BW69, Li84] imply that the same is true for search with delay greater than 0.

Our delay-d algorithm searches interval $[1, n]$ with $log_{\varphi_d} n + O(1)$ queries where φ_d is the positive root of $x^{d+1} = x^d + 1$. In particular, this gives $\varphi_1 \approx 1.618...$ (compared to $\psi_1 = \sqrt{2} \approx 1.415...$ for the unimodal problem) and $\varphi_2 \approx 1.466...$ (compared to $\psi_2 \approx 1.325...$). Our algorithm works for any delay d and we prove it optimal.

2 The main theorem

DEFINITION 3. *Let $A_d(t)$ denote the largest natural number such that the discrete monotone search problem with delay d on $[1, A_d(t)]$ can be solved with t queries.*

DEFINITION 4. *Let*

$$B_d(t) = \begin{cases} 0 & -d \leq t \leq 0 \\ B_d(t-1) + B_d(t-d-1) + 1 & 0 < t \end{cases}$$

THEOREM 2.1. *For all $d \geq 0$ and $t \geq 0$, $A_d(t) = B_d(t)$.*

Proof. Due to space constraints, we only give the proof for the upper bound.

We construct the algorithm inductively. If $t \leq d$, then $B_d(t) = t$ and we can just ask for the values of f at all integer points $1, 2, \ldots, B_d(t)$.

If $t > d$, we ask the first $d+1$ queries at points $B_d(t-1)+1$, $B_d(t-2)+1$, ..., $B_d(t-d-1)+1$. After receiving the answer $f(B_d(t-1)+1)$:

1. If $f(B_d(t-1)) \leq y_0$, then the root of $f(x_0) = y_0$ is in the interval $[B_d(t-1), B_d(t)]$. The size of this interval is $B_d(t) - B_d(t-1) - 1 = B_d(t-d-1)$ and we can apply the algorithm for searching with $t-d-1$ queries on it.

2. If $f(B_d(t-1)+1) > y_0$, then the root is in $[1, B_d(t-1)]$. We ask the $d+2^{\text{nd}}$ query at the point $B_d(t-d-2)+1$. Now, we are in a similar situation as we were before: we have the interval $[1, B_d(t-1)]$, $t-1$ queries and we have asked $d+1$ of them at the points $B_d(t-2)+1$, $B_d(t-3)+1$, ..., $B_d(t-d-2)+1$. We continue similarly.

The optimality is proved by generalizing $A_d(t)$ and $B_d(t)$ to include all situations that can appear in the search process, and using induction for the generalized A and B.

Resolving the recurrence of Definition 4, we get

COROLLARY 2.1. *Let φ_d be the positive real root of $x^{d+1} = x^d + 1$. Then $\log_{\varphi_d} n + O(1)$ queries are necessary and sufficient for the monotone search problem of size n with delay d.*

References

[Aig96] Martin Aigner. Searching with lies. *J. Comb. Th., A*, 74(1):43–56, 1996.

[AW66a] M. Avriel and D. J. Wilde. Optimal search for a maximum with sequences of simultaneous function evaluations. *Management Science*, 12:722–731, 1966.

[BW69] John H. Beamer and Douglass J. Wilde. Time delay in minimax optimization of unimodal functions of one variable. *Management Science*, 15:528–538, 1969.

[DGW92] Aditi Dhagat, Peter Gacs, and Peter Winkler. On playing 'twenty questions' with a liar. In Greg Frederickson, editor, *Proc.* 3rd *ACM-SIAM SODA*, 1992.

[GR93] Arthur S. Goldstein and Edward M. Reingold. A Fibonacci version of Kraft's inequality applied to discrete unimodal search. *SIAM J. Comp.*, 22(4):751–777, 1993.

[Guz90] Wojciech Guzicki. Ulam's searching game with two lies. *J. Comb. Th., A*, 54(1):1–19, 1990.

[Kie53] J. Kiefer. Sequential minimax search for a maximum. *Proc. AMS*, 4:502–505, 1953.

[KM68] Richard M. Karp and Willard L. Miranker. Parallel minimax search for a maximum. *J. Comb. Th.*, 4(1):19–35, 1968.

[Li84] Weixuan Li. *Optimal sequential block search*, Heldermann Verlag Berlin, 1984.

[MR96] Anmol Mathur and Edward M. Reingold. Generalized Kraft's inequality and discrete k-modal search. *SIAM J. Comp.*, 25(2):420–447, 1996.

[OW64] L. T. Oliver and D. J. Wilde. Symmetrical sequential minmax search for a maximum. *Fibonacci Quarterly*, 2:169–175, 1964.

[Pel87] Andrzej Pelc. Solution of Ulam's problem on searching with a lie. *J. Comb. Th., A*, 44(1):129–140, 1987.

[Pel89] Andrzej Pelc. Detecting errors in searching games. *J. Comb. Th., A*, 51(1):43–54, 1989.

[Ula76] S. M. Ulam. *Adventures of a Mathematician.* Scribner's, New York, 1976.