References

[AD91] Javed Aslam and Aditi Dhagat. Searching in the presence of linearly bounded errors. To Appear In STOC, 1991.

[Pel89] Andrzej Pelc. Searching wih known error probability. *Theoretical Computer Science*, 63:185–202, 1989.

[RMK+80] R. L. Rivest, A. R. Meyer, D. J. Kleitman, K. Winklmann, and J. Spencer. Coping with errors in binary search procedures. *Journal of Computer and System Sciences*, 20:396–404, 1980.

[SW90] Joel Spencer and Peter Winkler. Three thresholds for a liar. Preprint, 1990.

into pairs $P_1, \ldots, P_m$, where P_j is the pair of answers associated with the ends of interval I_j. The set T is the disjoint union of the pairs P_j.

Paul asks two questions in a stage. First, he asks the question referring to the midpoint of I_m. Without loss of generality, suppose that the answer a_0 says x is smaller than the midpoint of I_m. Then, next he asks a question referring to the left endpoint of I_m. If the answer b_1 is consistent with P_m, then he removes a_0 and adds it to b_1 to form P_{m+1}. If b_1 contradicts P_m (this never happens if $m = -1$) then he removes P_m from the trusted set and puts it, together with a_0 and b_1, as a contradictory 4-tuple, into the discarded set.

A computation completely analogous to the case of tree questions shows that this strategy is successful.

To achieve the bound 1/3, the sequence $I_0, \ldots, I_m$ of intervals must decrease slower: each binary interval occurring in it must occur

$$s = \left\lceil \frac{2}{1 - 3r} \right\rceil$$

times. With a careful strategy, Paul discards contradicting 3-tuples rather than 4-tuples most of the time, which results in the improved bound 1/3. (Only about every sth discarded tuple is a 4-tuple.)

4 The Prefix-Bounded Game

Now we consider the game where Carole must adhere to the fraction r throughout the game. That is, during any initial set of i questions, Carole may lie at most $\lfloor ri \rfloor$ times. Aslam and Dhagat [AD91] have considered this problem when Paul may only ask cut questions and shown the following:

Theorem 6 (Aslam, Dhagat) *For any $r < 1/2$, Paul has a winning strategy in the prefix-bounded game played in $\{1, \cdots, N\}$ of $O(N^{\log_2(\frac{1}{1-r})})$ cut questions.*

Since the exponent of the bound is always smaller than 1 when $r < 1/2$, the bound is sub-linear.

Paul's strategy to achieve this bound is simple: he does a binary search using cut questions, but with the following modification. He maintains a set of "candidates" for x, and associated with each candidate is a number of lies that have been told if that candidate is the true value of x. After q questions have been asked, a candidate is thrown out if the number of lies associated with it is greater than $\lfloor rq \rfloor$. Since each new question of the binary search can be picked so as to divide the remaining candidates in half, it is asked repeatedly until one or the other half of the candidates can be thrown out. Then a new question is picked to divide the remaining candidates in half.The analysis of this strategy to show the above bound can be found in Aslam and Dhagat's paper [AD91].

This strategy can easily be extended to work with bit questions. Each successive bit question, starting with one about the first bit, cuts the pool of candidates by a half. Thus a search which asks successive bit questions repeatedly until one or the other half of the remaining candidates are eliminated is essentially following the same strategy as the one above for cut questions. Thus we can conclude that:

Corollary 1 *For any $r < 1/2$, Paul has a winning strategy in the prefix-bounded game played in $\{1, \cdots, N\}$ of $O(N^{\log_2(\frac{1}{1-r})})$ bit questions.*

Tree Questions. Let us give the proof first for the tree question case. For simplicity, let us assume that N is a power of two, with

$$n = \log N.$$

The strategy consists of *stages*, where each stage consists of one or two questions. Between any two stages, Paul divides the answers into the set T of trusted and the set D of discarded answers. The set D of discarded answers is the union of inconsistent subsets where each subset contains two or three answers. The trusted set T comes from a sequence of positive answers giving rise to the corresponding nested binary question intervals $I_1, \ldots, I_m$, where for all $j \leq n$, the length of I_j is $N/2^j$.

We describe the questions leading from one stage to the next one. Suppose first that I_m has length greater than 1 and it corresponds to answer a_0. Let us denote the left and right halves of I_m by U and V. Then in the first question of this stage, Paul asks about the set U. If the answer a_1 is positive, then he adds it to the set of trusted answers and the stage ends here. If it is negative, then Paul also asks V. If this answer a_2 is positive, then he adds it to the set of trusted answers and the stage ends here. If it is negative, then he removes a_0 from the trusted answers and adds the contradictory triple $\{a_0, a_1, a_2\}$ to the discarded answer set D.

Suppose now that I_m has length 1. Then Paul asks I_m again; let this be a_1. If the answer is positive, he adds it to the trusted answers. If it is negative, he removes a_0 from the trusted answers and adds the contradictory pair $\{a_0, a_1\}$ to the discarded answers.

Let $Q = \left\lceil \frac{n}{1-3r} \right\rceil$ be the number of questions asked. Let us take a count at the end of the game. The (not necessarily integer) number $d = |D|/3$ is at least as large as the number of false answers, therefore

$$\begin{aligned} d &\leq rQ, \\ Q &= |T| + 3d \leq |T| + 3rQ, \\ |T| &\geq Q(1-3r) \geq n. \end{aligned}$$

It follows that I_m has length 1. Let $u = |T| - (n-1)$. Then $u \geq 1$ and the last u intervals I_j are identical and have length 1. Let us show that x is in this interval. Suppose it is not. Then the last interval is false. But then at least $u + d$ answers are false: d in the discarded set and the last u trusted answers. We have (remember that $u \geq 1$)

$$\begin{aligned} n + \frac{3rn}{1-3r} &= \frac{n}{1-3r} \\ &\leq Q = n + u + 3d - 1 = n + 3(u+d) - 2u - 1 \leq n + 3(u+d) - 3, \end{aligned}$$

hence

$$\frac{rn}{1-3r} + 1 \leq u + d \leq rQ \leq \frac{rn}{1-3r} + r.$$

This contradiction proves that x is in the last trusted interval (which has length 1).

Cut Questions. In case of cut questions, a similar strategy is used. Rather than give the strategy for $r < 1/3$, we will give first a weaker one working for $r < 1/4$ and giving

$$Q = \left\lceil \frac{2(\log N + 1)}{1-4r} \right\rceil.$$

There is again a sequence $I_0, \ldots, I_m$ of nested trusted binary intervals and a set D of discarded questions that consists of contradictory 3-tuples and 4-tuples. Paul will group all trusted answers

($\Longleftarrow$) Again letting $M = \lceil \log N \rceil$ be the number of different bit questions that Paul can ask, we note that $Q \geq (\lfloor rQ \rfloor + 1)M + \lfloor rQ \rfloor$. Thus, we will be done if we can show a winning strategy for Paul which asks no more than $(\lfloor rQ \rfloor + 1)M + \lfloor rQ \rfloor$ questions.

If a bit question is not answered consistently, then we know that the number of lies told about it is at least the minimum of the number of "yes" answers and the number of "no" answers. If, at any point during the questioning, this minimum becomes larger than $\lfloor rQ \rfloor$, then we can stop asking that bit question and determine the true value of that bit. So Paul's winning strategy will be as follows:

Phase 1: Ask each bit question $\lfloor rQ \rfloor + 1$ times, keeping track of the $l_i = \min(Y_i, N_i)$, for each bit i, where Y_i is the number of "yes" answers given in response to the question about bit i and N_i is the number of "no" answers.

Phase 2: For those bits i which have $l_i \leq \lfloor rQ \rfloor$, ask the i-th bit question until $\max(Y_i, N_i) = \lfloor rQ \rfloor + 1$.

During phase 1, Paul asks exactly $(\lfloor rQ \rfloor + 1)M$ questions.

Claim 1 *Paul asks at most $\lfloor rQ \rfloor$ questions in phase 2.*

Proof: At the end of the game, for any bit i, $\max(Y_i, N_i) \leq \lfloor rQ \rfloor + 1$. So:

$$\sum_{i=1}^{M} \max(Y_i, N_i) \leq M(\lfloor rQ \rfloor + 1).$$

We also know that:

$$\sum_{i=1}^{M} l_i \leq \lfloor rQ \rfloor$$

Since $Q = \sum_{i=1}^{M} [\max(Y_i, N_i) + l_i]$, it must be that $Q \leq M(\lfloor rQ \rfloor + 1) + \lfloor rQ \rfloor$. Since $M(\lfloor rQ \rfloor + 1)$ of these questions were asked in phase 1, at most $\lfloor rQ \rfloor$ questions must have been asked in phase 2. ■

3.2 Cut and Tree Questions

In this subsection, it will be more convenient to say that the numbers range between 0 and $N - 1$.

We will investigate two possible sets of allowed questions: *cut questions*, i.e., when questions refer to intervals of the form $\{0, \ldots, b-1\}$ for all b and *tree questions*, when they refer to *binary* intervals, i.e., intervals of the form $[i2^j, (i+1)2^j)$ for all i, j. It turns out that in both cases, there is a strategy for the Paul if and only if $r < 1/3$ and with $Q = O(\log N)$. The fact that Carole has a winning strategy here when $r \geq 1/3$ follows from the fact that Carole has winning strategy for $r \geq 1/3$ even when Paul may ask general questions.

Theorem 5
With tree questions, Paul has a winning strategy for all $r < 1/3$ asking

$$Q = K_{\text{tree}}(r, N) = \left\lceil \frac{\log N + 1}{1 - 3r} \right\rceil \textit{ questions.}$$

For cut questions, and $r < 1/3$, the bound is

$$Q = K_{\text{cut}}(r, N) = \left\lceil \frac{8 \log N}{(1 - 3r)^2} \right\rceil.$$

2.2 Cut Questions

Cut questions seem (somewhat counter-intuitively) to be even weaker than bit questions in the batch game. It turns out that Paul has a winning strategy here only for N less than about $\frac{1}{2r}$.

Theorem 3 *Paul has a winning strategy with Q cut questions in the batch game iff $N \leq \frac{Q}{2\lfloor rQ \rfloor + 1} + 1$.*

Proof:

($\Longleftarrow$) If $N \leq \frac{Q}{2\lfloor rQ \rfloor+1} + 1$, then Paul can win by asking each of the $N - 1 \leq \frac{Q}{2\lfloor rQ \rfloor+1}$ questions $(2\lfloor rQ \rfloor + 1)$ times, and determining the correct answer to each question by choosing the majority answer.

($\Longrightarrow$) Now there must be at least one cut question which is asked at most $2\lfloor rQ \rfloor$ times. Let us say this question is "Is x less than k?" Now Carole's winning strategy is as follows:

- Answer all questions of the type "Is x less than j?" where $j < k$ with a "yes".
- Answer all questions of the type "Is x less than j?" where $j > k$ with a "no".
- Answer half of the questions "Is x less than k?" with a "yes" and the other half with "no".

Then both k and $k + 1$ are possible values of x since both have been lied about at most $\lfloor rQ \rfloor$ times.

∎

3 The Adaptive Game

The adaptive game gives Paul greater power by allowing him to look at Carole's previous answer before he asks the next question. However, we know from Theorem 1 that even with general questions, Paul has no hope of winning when $r \geq 1/3$. We show here that, as in the batch game, bit questions are not powerful enough to allow Paul to win for arbitrarily large N. However, with cut questions, Paul has a winning strategy for all $r < 1/3$ which asks $O(\log N)$ questions.

3.1 Bit Questions

Theorem 4 *With Q bit questions, Paul has a winning strategy iff $N \leq 2^{\left\lfloor \frac{Q - \lfloor rQ \rfloor}{\lfloor rQ \rfloor + 1} \right\rfloor}$.*

Proof:

($\Longrightarrow$) Let $M = \lceil \log_2 N \rceil$ be the number of different bit questions that Paul can ask. Note that $N > 2^{(Q - \lfloor rQ \rfloor)/(\lfloor rQ \rfloor + 1)}$ iff $Q < (\lfloor rQ \rfloor + 1)M + \lfloor rQ \rfloor$.

Carole's winning strategy is as follows: she answers all of Paul's initial questions with a "no" until answering the next question would allow Paul to win the game. Note that Paul must have asked at least $(\lfloor rQ \rfloor + 1)M - 1$ questions during this initial phase, since each bit must be asked about at least $\lfloor rQ \rfloor + 1$ times in order for Paul to win the game with all "no's". Thus, there exists a bit i which has been asked about at most $\lfloor rQ \rfloor$ times. In the remaining questions, Carol continues to answer "no" to all questions except the ones about bit i, to which she answers "yes". Since there are at most $\lfloor rQ \rfloor$ questions left to be asked during this latter phase, both $000\ldots\underbrace{1}_{i}\ldots000$ and $000\ldots\underbrace{0}_{i}\ldots000$ will be possible values of x.

In this paper, we prove theorems of this nature when Paul is restricted in the sort of questions he may ask. In particular, we look at the classes of bit and cut questions and examine their strength against the three different (batch, adaptive and prefix-bounded) versions of this game.

We start in section 2 by considering the batch game. We show that the restricted question classes are too weak for Paul to win for arbitrarily large N. In section 3, we consider the adaptive game, and show that while bit questions are too weak for this model, Paul *can* win with $O(\log N)$ cut questions, for all $r < 1/3$. This latter bound is first shown for a slightly weaker class of questions, namely tree questions. Tree questions ask about an initial sequence of bits of the binary representation of x; thus they can be formed using a conjunction of bit questions. Since Paul has a winning strategy in the adaptive game with tree questions, but not with bit questions, we have some evidence that, given a certain primitive class of questions, it is advantageous to ask questions formed as conjunctions of primitives rather than asking the primitives themselves. Finally in section 4, following some results for cut questions in Aslam and Dhagat's paper [AD91], we show how Paul can win in the prefix-bounded game with restricted types of questions for any $r < 1/2$. Here, however, $O(\log N)$ questions seem no longer sufficient, as they are when Paul may ask general "yes-no" questions.

2 The Batch Game

In this game, Paul must submit all of his Q questions to Carole initially. Carole then sends Paul her answers to these questions, lying in at most $\lfloor rQ \rfloor$ answers.

2.1 Bit Questions

We show that Paul has a winning strategy only for small values of N, roughly for $N < 2^{\frac{1}{2r}}$. The precise bound is as follows.

Theorem 2 *Paul has a winning strategy with Q bit questions in the batch game iff $N \leq 2^{\left\lfloor \frac{Q}{2\lfloor rQ \rfloor + 1} \right\rfloor}$.*

Proof:

($\Longrightarrow$) Suppose $N > 2^{\left\lfloor \frac{Q}{2\lfloor rQ \rfloor + 1} \right\rfloor}$. Then the number of different bit questions available is $\lceil \log_2 N \rceil > \left\lfloor \frac{Q}{2\lfloor rQ \rfloor + 1} \right\rfloor$. Thus there exists a bit question, say about bit j, asked by Paul at most $2\lfloor rQ \rfloor$ times. Since Carol is allowed to see Paul's questions before answering them, she can win by the following strategy:

- Answer each question not about bit j with a "no" (i.e. tell Paul that these bits are all 0's).
- Answer half of the questions about bit j with a "yes" and the other half with a "no".

Then both $000\ldots\underbrace{1}_{j}\ldots000$ and $000\ldots\underbrace{0}_{j}\ldots000$ are possible values of x, since each has been lied about at most $\lfloor rQ \rfloor$ times. Note that both numbers are in the required range.

($\Longleftarrow$) Since the number of different bit questions is now at most $\left\lfloor \frac{Q}{2\lfloor rQ \rfloor + 1} \right\rfloor$, Paul can ask each bit question $(2\lfloor rQ \rfloor + 1)$ times, enabling him to determine the correct answer each time.

∎

1. **Bit Questions:** "Is the i-th bit in the binary representation of x equal to 1?"
2. **Cut (or Comparison) Questions:** "Is x less than y?", for some $y \in \{0, \cdots, N-1\}$.
3. **General (or Membership) Questions:** "Is $x \in S$?", where S is some subset of $\{0, \cdots, N-1\}$.

How and where Carole is allowed to place her lies also makes a difference. Three successively more restrictive models of the game will be considered:

1. **Batch:** Paul submits all his questions in batch. Thus, Carole is allowed to see all of Paul's questions before answering them.
2. **Adaptive:** Carole must answer Paul's questions on-line.
3. **Prefix-bounded:** Carole must answer Paul's questions on-line as well as never lie more than $\lfloor ri \rfloor$ times in any initial i questions.

This game is motivated from the problem of searching in a discrete bounded domain in the presence of malicious errors. In our game, x is the searched element and Carole is the malicious adversary introducing errors. An adversary who may lie arbitrarily is sure to prevent us from succeeding in the search. Thus, in the literature on this subject, the adversary is usually restricted in the number of lies she is allowed.

In their 1980 paper, Rivest, et al. [RMK$^+$80] considered the case of a constant number of lies. They showed that if there are no more than k lies, then $\log N + k \log\log N + O(k \log k)$ questions are sufficient to search. Recent work [Pel89, AD91, SW90] has concentrated on the case where the number of lies is proportional to the number of questions. In particular, Spencer and Winkler [SW90] have showed the following about a Paul who may ask general "yes-or-no" questions:

Theorem 1 (Spencer,Winkler)

1. *In the batch game on the set* $\{0, \cdots, N-1\}$*:*
   (a) *If* $r < 1/4$*, then Paul has a winning strategy with* $\Theta(\log N)$ *questions.*
   (b) *If* $r = 1/4$*, then Paul has a winning strategy with* $\Theta(N)$ *questions.*
   (c) *If* $r > 1/4$*, then Carole has a winning strategy for all* $N > N(r)$*, no matter how large* Q *may be.*
2. *In the adaptive game on the set* $\{0, \cdots, N-1\}$*:*
   (a) *If* $r < 1/3$*, then Paul has a winning strategy with* $\Theta(\log N)$ *questions.*
   (b) *If* $r \geq 1/3$*, then Carole has a winning strategy (for all* $N \geq 5$*).*
3. *In the prefix-bounded game on the set* $\{0, \cdots, N-1\}$*:*
   (a) *If* $r < 1/2$*, then Paul has a winning strategy with* $\Theta(\log N)$ *questions.*[1]
   (b) *If* $r \geq 1/2$*, then Carole has a winning strategy (for all* $N \geq 3$*).*

[1]This result has also appeared in a paper by Aslam and Dhagat [AD91], who present a slightly different proof.

On Playing "Twenty Questions" with a Liar

(Extended Abstract)

Aditi Dhagat*
Department of Mathematics
Massachusetts Institute of Technology
Cambridge, MA 02139

Peter Gács†
Department of Computer Science
Boston University
Boston, MA 02215

Peter Winkler‡
Bellcore 2L335
Morristown, NJ 07962-1910

April 16, 1999

Abstract

We consider a version of the game "Twenty Questions" played on the set $\{0, \cdots, N-1\}$ where the player giving answers may lie in her answers. The questioner is allowed Q questions and the responder may lie in upto $\lfloor rQ \rfloor$ of the answers, for some fixed and previously known fraction r. Under three different models of this game and for two different question classes, we give precise conditions (i.e. tight bounds on r and, in most cases, optimal bounds on Q) under which the questioner has a winning strategy in the game. The games are motivated by the problem of searching under various restricted question classes in the presence of errors.

1 Introduction

Consider the following version of the classic game "Twenty Questions." There are two players: Paul (a.k.a. the Questioner) and Carole (a.k.a. the Oracle). Carole thinks of a number x between 0 and $N-1$. Paul is permitted to ask Q "yes-or-no" questions, by means of which he must determine x. It is easy to see that Paul has winning strategy in this game if and only if N is no bigger than 2^Q. We make the game more interesting by allowing Carole to lie; in particular, in our game, Paul must determine x in Q questions where Carole may tell at most $\lfloor rQ \rfloor$ lies, for some fixed and previously known fraction r. Now when does Paul have a winning strategy?

The answer, of course, depends on how big r and Q are, on where Carole is allowed to place the lies, and on the kinds of questions Paul is allowed to ask. We will consider the following kinds of questions:

*Author was supported by DARPA Contract N00014-87-K-825, National Science Foundation Grant CCR-8912586, and Air Force Contract AFOSR-89-0271. Author's net address: aditi@theory.lcs.mit.edu

†Supported in part by NSF Grant CCR-9002614. Author's net address: gacs@cs.bu.edu

‡Author is on leave from Emory University; research at Emory supported by ONR grant N00014 85-K-0769. Author's net address: pw@bellcore.com