The Interval Liar Game

Benjamin Doerr[1], Johannes Lengler[2], and David Steurer[3]

[1] Max–Planck–Institut für Informatik, Saarbrücken, Germany.
[2] Mathematics Department, Saarland University, Saarbrücken. johnny@math.uni-sb.de
[3] Computer Science Department, Princeton University, Princeton.
dsteurer@cs.princeton.edu

Abstract. We regard the problem of communication in the presence of faulty transmissions. In contrast to the classical works in this area, we assume some structure on the times when the faults occur. More realistic seems the "burst error model", in which all faults occur in some small time interval.

Like previous work, our problem can best be modelled as a two-player perfect information game, in which one player ("Paul") has to guess a number x from $\{1, \ldots, n\}$ using Yes/No-questions, which the second player ("Carole") has to answer truthfully apart from few lies. In our setting, all lies have to be in a consecutive set of k rounds.

We show that (for big n) Paul needs roughly $\log n + \log\log n + k$ rounds to determine the number, which is only k more than the case of just one single lie.

1 Introduction and Results

Communication in the presence of transmission faults is a well-studied subject. Pelc's [Pel02] great survey lists more than a hundred references on such problems.

1.1 Communication Model with Errors

The customary model is that there are two entities, "Sender" and "Receiver". Sender wants to send a message to Receiver. The message is represented by a number x from $[n] := \{1, \ldots, n\}$. If we have an error-free channel, it is clear that Sender needs to send $\log(n) := \log_2(n)$ bits (and Receiver only needs to listen).

In the model with errors, however, some of the bits sent by Sender are flipped. Of course, we need some restriction on the occurrence of errors, as otherwise no reliable communication is possible. Typically, we assume that such errors only occur a certain number of times, at a certain rate or according to a certain probability distribution.

To compete with the errors, we often assume a two-way communication, that is, Receiver may send out information to Sender. However, we typically think of the situation as not symmetric: Bits sent from Receiver to Sender are never flipped (no errors occur). This model is justified in many practical situations where one communication partner has much less energy available and thus his sendings are more vulnerable to errors.

1.2 Liar Games

We often adopt a worst-case view. Hence we do not assume the errors to be random, but rather to be decided on by a malevolent adversary. In fact, we may think of that sender not really wanting to share his secret x, but rather trying to keep it by intentionally causing errors (lying). This leads to a so-called *liar game*. In the following, we adopt the language usually used in the analysis of such games. In particular, Sender/Lier will be called "Carole", an anagram of oracle, and Receiver, who is questioning Carole to reveal the secret, will be called "Paul" in honor of Paul Erdős, the great questioner.

The rules of the game are as follows: Carole decides on a number (secret) $x \in [n]$. There are q rounds. Each round, Paul asks a Yes/No-question, which Carole answers. In doing so, Carole may lie according to further specifications. Paul wins the game, if after q such rounds, he knows the number.

To make this a perfect information game (in-line with our worst-case view), let us assume that Carole does not have to decide on the number x beforehand, but rather tries to answer in a way that is consistent with some secret. For technical reasons, we shall also allow that she lies in a way that is inconsistent with any secret, which will be viewed as a win for Paul as well.

We remark that, depending on the parameters n, q, and on the lying restrictions either Paul or Carole has a winning strategy. So we say that Paul wins if he has a winning strategy.

Note that this set-up perfectly models the communication problem with errors. There is one more remark regarding Paul's questions. It seems that his communication effort is much higher, since each question can only be represented by a n bit string.

This could be justfied by the stronger battery Paul has compared to Carole, but there is a more natural explanation: If Paul and Carole agree on a communication protocol beforehand, then Paul does not need to transmit his questions. It suffices that he merely repeats the bit he just received and Carole can deduce the next question from this and the agreed-on protocol.

In the following, we rather use the language of games than that of communication protocols. With the above equivalence at hand, this is merely a question of taste and we follow the authors of previous work in this respect.

1.3 Previous Results

As said, liar games are an intensively studied subject. We now briefly state the main results relevant for our work and refer to the survey paper Pelc [Pel02] for a more complete coverage.

The first to notice the connection between erroneous communication and such games was Alfréd Rényi [Rén61,Rén76]. However, for a long time most of this community was not aware of Rényi's work and cited Ulam [Ula76] as inventor of liar games.

Pelc [Pel87] was the first to completely analyse the game with one lie. He showed that Paul wins for even n if $n \leq 2^q/(q+1)$, and for odd n if $n \leq (2^q - q + 1)/(q+1)$. There are numerous results for $k = 2$, 3, or 4 lies, which we will not discuss here.

Spencer [Spe92] solved the general problem for any fixed number k of lies. Here Paul wins if $n \leq 2^q / \binom{q}{\leq k} (1 + o(1))$, where $\binom{q}{\leq k} = \sum_{i=0}^{k} \binom{q}{i}$.

All results above concern the fully adaptive ('real game') setting with unrestricted questions and fixed numbers of lies. The problem has a quite different nature if only comparison questions ("Is $x \leq s$?" for some $s \in [n]$) are allowed [BK93], a constant fraction of lies is allowed in any initial segment of rounds [Pel89], or Paul's questions have to come in two batches, where Carole gives her answers only after having received the whole batch [CM99].

1.4 Our Contribution

Translating the above results back into the model of erroneous communication, the errors occur independently at arbitrary times. While this might be true for some types of errors, we feel that it is much more likely that the errors occur in bunchs. We think, e.g., of atmospheric disorders. Here, not only a single bit will be affected, but a whole sequence of bits sent.

In the game theoretic setting, we allow Carole to lie up to k times, but only in a way that all lies occur in k consecutive rounds. Note that, in these k rounds, Carole may lie, but of course she does not have to.

The additional interval restriction makes Carole's position much harder. Roughly speaking, Paul only needs k more questions than in the one-lie game. This shows that, in scenarios where it can be assumed, using our interval assumption is a valuable improvement. More precisely, we show the following.

Theorem 1. *Let $n, q \in \mathbb{N}$ and $k \in \mathbb{N}_{\geq 2}$.*

(i) Paul wins if $q \geq \lceil \log n \rceil + k + \lceil \log \log 2n \rceil$ and $q \geq \lceil \log n \rceil + 2k$.
(ii) Carole wins if $q < \log n + 2k$.
(iii) Carole wins if $q < \log n + k + \log \log 2n - 1$.

We assumed $k \geq 2$ as otherwise the game in consideration would revert to the searching game with just one lie.

Note that Theorem 1 gives almost matching lower and upper bounds on the number of questions Paul needs to reliably distinguish n integers. Specifically, for all choices of n and k, the upper and lower bound differ by at most 3.

2 Notation and Preliminaries

We describe a game position by a non-negative vector $P = (x_k, \ldots, x_0)$, where x_i is the number of integers for which (assuming it to be the correct answer) Carole is allowed to lie within the next i questions. Note that for the analysis, it does not matter which are the particular integers that Carole may lie for i times, it is only their number that matters.

In particular, x_k is the number of integers for which Carole has never lied, and x_0 is the number of integers for which Carole must not lie anymore. Note that $\sum_{i=0}^{k} x_i \leq n$, and this is strict if there are integers for which Carole would have lied at two times separated by at least k rounds. For the initial position, denoted P^0, we have $x_k = n$ and $x_0 = \ldots = x_{k-1} = 0$.

We continue formalizing the questions Paul is asking. Note first that a Yes/No-questions can always be expressed in the form “$x \in S$?” for some $S \subseteq [n]$. Since again for the analysis the particular integers are not so relevant, we describe the question via an integer vector $v = (v_k, \dots, v_0)$, where v_i is the number of integers that (i) are in S and (ii) Carole may lie i times for. Consequently, we have $0 \le v_i \le x_i$ for all $i \in \{0, \dots, k\}$. To ease the language, we identify questions with their corresponding vectors.

Depending on Carole’s answer there are two possibilities for the next game position P', namely $P' = \mathit{YES}(P, v)$ and $P' = \mathit{NO}(P, v)$, where

$$\begin{aligned} \mathit{YES}(P, v) &= (v_k, x_k - v_k, x_{k-1}, x_{k-2}, \dots, x_1 + v_0) \\ \mathit{NO}(P, v) &= \mathit{YES}(P, P - v) = (x_k - v_k, v_k, x_{k-1}, x_{k-2}, \dots, x_1 + x_0 - v_0) \end{aligned}$$

Note that neither $\mathit{YES}(P, v)$ nor $\mathit{NO}(P, v)$ depends on any v_i with $0 < i < k$. For the integers corresponding to these entries, Carole’s answer does not affect the state of the game.

For a position $P = (x_k, \dots, x_0)$, a question (i.e. an integer vector v with $0 \le v \le P$) is a *perfect bisection* if $v_0 = \frac{1}{2} x_0$ and $v_k = \frac{1}{2} v_k$.

Recall that $\mathit{YES}(P, v)$ and $\mathit{NO}(P, v)$ do not depend on $v_1, \dots, v_{k-1}$, so if Paul can make a perfect bisection, then the successor state does not depend on Carole’s answer.

We call a question a *quasi-perfect bisection* if $v_i \in \{\lfloor x_i/2 \rfloor, \lceil x_i/2 \rceil\}$ for $i = 0$ and $i = k$.

We conclude this section by explaining when some position is better than another:

Lemma 2. *Let $P = (x_k, \dots, x_0)$ and $P' = (x'_k, \dots, x'_0)$ be positions (= non-negative inegral vectors). Assume that P and P' have the following property:*

$$\sum_{i=j}^{k} x_i \le \sum_{i=j}^{k} x'_i \qquad \textit{for all } j = 0, \dots, k. \tag{1}$$

Then for any q, we have the implication

$$\textit{Paul can win } P' \textit{ in } q \textit{ rounds} \implies \textit{Paul can win } P \textit{ in } q \textit{ rounds}$$

In this case, we call position P *at least as good* as P', and we call P' *at most as good* as P.

Proof. Though the statement is rather technical, the idea is simple: We can generate P' out of P by *(i)* allowing Carole some additional lies and *(ii)* adding some more numbers to the search space. Clearly, both operations will make the game harder for Paul, so if he has a winning strategy for P' in q rounds, then exactly the same strategy will also win P.

So we want to prove that we can indeed transform P into P' by operations *(i)* and *(ii)*. We use an inductive argument. Firstly, we add some numbers of type x_0 to P until we get equality for $j = 0$, i.e., $\sum_{i=0}^{k} x_i = \sum_{i=0}^{k} x'_i$.

Now we have $x_0 = \sum_{i=0}^{k} x_i - \sum_{i=1}^{k} x_i \ge \sum_{i=0}^{k} x'_i - \sum_{i=1}^{k} x'_i = x'_0$, so $x_0 - x'_0 \ge 0$. We choose $x_0 - x'_0$ numbers in P at the x_0-position. For these numbers, we allow

Carole to lie in the next step. So we get a new position $P^1 = (x_k, \ldots, x_2, x_1 + x_0 - x'_0, x'_0)$, and we know that P is at least as good as P^1.

Now inductively we produce a sequence $P^0 := P, P^1, P^2, \ldots, P^k$ with the following properties:

- P^{i-1} is at least as good as P^i (in the sense of equation (1)).
- P^i is generated from P^{i-1} by operations of type *(i)* and *(ii)*.
- For $0 \le j < i$ we have $x^i_j = x'_j$, where x^i_j is the j-entry of P^i.
- $\sum_{j=0}^k x^i_j = \sum_{j=0}^k x'_j$ for $i > 0$.

Indeed, we have already constructed P^1. Out of P^{i-1}, by the same construction we get P^i, namely by allowing one additional lie for some numbers from x^{i-1}_i. (Formally by setting $P^i := (x^{i-1}_k, \ldots, x^{i-1}_{i+1}, x^{i-1}_i + x^{i-1}_{i-1} - x'_{i-1}, x'_{i-1}, x^{i-1}_{i-2}, \ldots, x^{i-1}_0)$). Note that P^{i-1} and P^i are identical except for the components $i-1$ and i. It is easy to check that P^i has the desired properties.

Finally, we end up with P^k, which is automatically identical to P'.

Altogether, we have constructed P' out of P by the feasible operations *(i)* and *(ii)*. This proves the claim.

3 Upper Bounds and Strategies for Paul

In this section, we give a strategy for Paul. In this way, we derive upper bounds on the number of questions Paul needs in order to reveal the secret $x \in [n]$. We show (Corollary 6) that for n being a power of 2, Paul can win if

$$q \ge \max\left\{k + \log n + \lceil \log\log n \rceil,\ 2k + \log n\right\}.$$

Our strategy is constructive, that is, immediately yields an efficiently executable protocol for the underlying communication problem.

Here is an outline of the strategy. Assume that n is a power of two. Clearly, some strategy working for a larger n will also work for a smaller one, hence this assumption is fine (apart from possible a minor loss in the resulting bounds). If all x_i are even, Paul can ask the question $v = \frac{1}{2}P$. He does so for the first $\log n$ rounds of the game (Main Game), resulting in a position with $x_k = 1$. Now the aim is to get rid of this one integer Carole has not lied for yet. To do so, we ask a "trigger question", roughly $(1, 0, \ldots, 0)$. Either we succeeded with our plan and simply repeat asking for half of the x_0-integers (Endgame I), or we end up with very few possible integers altogether (Endgame II), allowing an easy analysis.

Lemma 3 (Main Game). *If n is a power of 2, then with the first $m = \log n$ questions Paul can reach position*

$$P^m = (1, 1, 2, \ldots, 2^{k-2}, (m-k+1)2^{k-1}).$$

Proof. In the first m rounds, Paul can always ask questions of the form $v = P/2$, where P is the current game position. The position after k such perfect bisections is

$$P^k = (2^{m-k}, 2^{m-k}, 2^{m-k+1}, \ldots, 2^{m-1}).$$

A simple inductive argument shows that the position after $k+\nu$ questions with $\nu \leq m-k$ is

$$P^{k+\nu} = (2^{m-k-\nu}, 2^{m-k-\nu}, 2^{m-k-\nu+1}, \dots, 2^{m-\nu-2}, (\nu+1)\cdot 2^{m-\nu-1}).$$

For $\nu = m-k$, we get the statement of the lemma.

After the first m questions, Paul asks a "trigger question" $v^{m+1} = (1, 0, \dots, 0, 2^{k-2})$. If k is sufficiently small compared to n, Carole will not give up the relatively many possibilities encoded in x_0 and therefore answer "No". The following two lemmas deal with both possible successor positions, namely $\mathit{YES}(P^m, v^{m+1})$ and $\mathit{NO}(P^m, v^{m+1})$.

Lemma 4 (Endgame I). *From position*

$$\mathit{NO}(P^m, v^{m+1}) = (0, 2^0, 2^0, \dots, 2^{k-3}, (m-k+1)2^{k-1})$$

Paul wins the game (by reaching position $(0, \dots, 0, 1)$*), with at most* $k-1+\lceil \log m \rceil$ *questions.*

Proof. With $k-2$ perfect bisections, Paul reaches the position with $x_k = \dots = x_2 = 0$, $x_1 = 1$ and $x_0 = 2(m-k+1) + \sum_{i=1}^{k-2} 2^{i-1}/2^{i-1} = 2m-k$.

In the next question, Paul asks for $m - \lceil k/2 \rceil$ integers corresponding to the last entry of the position. So the next position is no more than

$$(0, \dots, 0, m - \lfloor k/2 \rfloor + 1) \leq (0, \dots, 0, m).$$

From this position on, the game reverts to classical "Twenty Questions" problem for a universe of size m. So Paul can win with $\lceil \log m \rceil$ additional questions.

The total number of questions is at most

$$k-2\ +1\ +\lceil \log m \rceil \leq k-1+\lceil \log m \rceil.$$

Lemma 5 (Endgame II). *Paul can win with at most* $2k-1$ *questions from position*

$$\mathit{YES}(P^m, v^{m+1}) = (1, 0, 2^0, 2^1, \dots, 2^{k-3}, 2^{k-2})$$

Proof. With $k-2$ quasi-perfect bisections, Paul reaches a position at least as good as

$$(1, 0, \dots, 0, \sum_{i=0}^{k-2} 2^i/2^i) = (1, 0 \dots, 0, k-1).$$

Now Paul asks for the number corresponding to the first entry of the position, that is, the question $v = (1, 0, \dots, 0)$. If the answer is "Yes", Paul wins instantly. Otherwise, the position is $(0, 1, 0, \dots, k-1)$. Playing the "Twenty Questions" game on the $k-1$ integers corresponding to the last entry, we reach with $t \leq \lceil \log k \rceil$ additional questions a position with $x_0 = x_{k-1-t} = 1$ and all other entries naught. From this position, Paul can win in $k-t$ questions.

The total number of questions is at most

$$k-2\ +\ 1\ +\ k = 2k-1.$$

Corollary 6. *For $\log n \in \mathbb{N}$, Paul can win if*

$$q \geq \max\left\{k + \log n + \lceil \log\log n \rceil,\ 2k + \log n\right\}.$$

Proof. By Lemma 3, we need $\log n$ questions for the main game. Then Paul asks one "trigger question". Depending on Carole's answer, Paul either plays Endgame I or Endgame II. In the first case, he needs $k + \lceil \log\log n \rceil - 1$ further questions to win the game (Lemma 4). In the latter case, Paul wins with $2k - 1$ questions (Lemma 5).

If n is not a power of two, we can replace the starting position $P = (n, 0, \ldots, 0)$ by $(2^{\lceil \log(n) \rceil}), 0, \ldots, 0)$, which is at most as good as P. By the Corollary, Paul can still win if

$$q \geq \max\left\{k + \lceil \log n \rceil + \lceil \log\log n \rceil,\ 2k + \lceil \log n \rceil\right\},$$

which is the statement in Theorem 1 (i).

4 Lower Bound

In this section, we prove lower bounds showing that our strategies given in the previous section are optimal up to a small constant number of questions. We start by defining the following *formal weight function*:

$$w_j(x_k, \ldots, x_0) = (j - k + 2)2^{k-1}x_k + \sum_{i=0}^{k-1} 2^i x_i.$$

The weight function is supposed to determine whether it is possible for Paul to find out the correct number in j rounds. It does not quite so, but it solves only a formal relaxation of the problem. (That's why it is called *formal* weight function.)

Note that the weight function is linear in its variables.

The following lemma summarises the important properties of such a formal weight function.

Lemma 7. *(i)* ***Triangle equality:*** *For all $j \geq k + 1$ and for all integral vectors P and v,*

$$w_j(P) = w_{j-1}(\mathit{YES}(P, v)) + w_{j-1}(\mathit{NO}(P, v)).$$

(Note: We do not require that the entries of P and v are positive.)

(ii) ***Formal descent:*** *For all $j \geq k + 1$ and for all integral P, there is a formal choice v for Paul, such that*

$$w_{j-1}(\mathit{YES}(P, v)) = w_{j-1}(\mathit{NO}(P, v)), \text{ if } w_j(P) \text{ is even.}$$

$$w_{j-1}(\mathit{YES}(P, v)) = w_{j-1}(\mathit{NO}(P, v)) + 1, \text{ if } w_j(P) \text{ is odd.}$$

By a formal choice, we mean an integral vector with possibly negative entries.

(iii) ***Starting condition:*** *For $j = k$, if P is a state with non-negative integral entires, we have $w_k(P) \leq 2^k$ if and only if Paul can win the situation P in k rounds.*

Proof. Let $P = (x_k, \dots, x_0)$, $v = (v_k, \dots v_0)$. Direct calculation proves the assertion:

$$
\begin{aligned}
& w_{j-1}(\mathit{YES}(P,v)) + w_{j-1}(\mathit{NO}(P,v)) \\
= & \left((j-k+1)2^{k-1}v_k + 2^{k-1}(x_k - v_k) + \sum_{i=1}^{k-2} 2^i x_{i+1} + (v_0 + x_1) \right) \\
& + \left((j-k+1)2^{k-1}(x_k - v_k) + 2^{k-1}v_k + \sum_{i=1}^{k-2} 2^i x_{i+1} + (x_0 - v_0 + x_1) \right) \\
= & (j-k+1)2^{k-1}x_k + 2^{k-1}x_k + \sum_{i=1}^{k-2} 2^{i+1} x_{i+1} + 2x_1 + x_0 \\
= & (j-k+1)2^{k-1}x_k + \sum_{i=2}^{k-1} 2^i x_i + 2x_1 + x_0 \\
= & (j-k+1)2^{k-1}x_k + \sum_{i=0}^{k-1} 2^i x_i \\
= & w_j(P)
\end{aligned}
$$

This proves the triangle equality.

Obviously, if $P = (0, \dots, 0, 2, 0, \dots, 0)$, then Paul can choose $v = (0, \dots, 0, 1, 0, \dots, 0)$, and thus obtain $w_{j-1}(\mathit{YES}(P,v)) = w_{j-1}(\mathit{NO}(P,v))$. (Because by symmetry $\mathit{YES}(P,v) = \mathit{NO}(P,v)$.)

But w_j is linear in all entries, so it suffices to prove the claim for $P = (0, \dots, 0, 1, 0, \dots, 0)$, with the i-th entry $= 1$. Let $P' = (0, \dots, 0, 1, 0, \dots, 0)$, but with the $i-1$-th entry $= 1$. Now put $a := w_j(P) - 2w_{j-1}(P')$. We must distinguish two cases:

- $w_j(P)$ is even: Then also a is even. Put $v := P + (0, \dots, 0, \frac{a}{2})$. Then $\mathit{YES}(P,v) = P' + (0, \dots, 0, \frac{a}{2})$, so $w_{j-1}(\mathit{YES}(P,v)) = w_{j-1}(P') + \frac{a}{2} = \frac{1}{2}w_j(P)$. On the other hand, by the triangle equality, $w_{j-1}(\mathit{NO}(P,v)) = w_j(P) - w_{j-1}(\mathit{YES}(P,v)) = \frac{1}{2}w_j(P) = w_{j-1}(\mathit{YES}(P,v))$.
- $w_j(P)$ is odd: Then also a is odd. Put $v := P + (0, \dots, 0, \frac{a+1}{2})$. Then $\mathit{YES}(P,v) = P' + (0, \dots, 0, \frac{a+1}{2})$, so $w_{j-1}(\mathit{YES}(P,v)) = w_{j-1}(P') + \frac{a+1}{2} = \frac{1}{2}(w_j(P) + 1)$. On the other hand, by the triangle equality, $w_{j-1}(\mathit{NO}(P,v)) = w_j(P) - w_{j-1}(\mathit{YES}(P,v)) = \frac{1}{2}(w_j(P) - 1) = w_{j-1}(\mathit{YES}(P,v)) - 1$.

For the starting condition, note that due to $j = k$, the weight function simplifies to $w_k(x_k, \dots, x_0) = \sum_{i=0}^{k} 2^i x_i$.

Case 1: $x_k \geq 1$

In this case, there is a chip C_1 on the x_k-position.

First assume that the weight is $\geq 2^k$. Then there is some other chip C_2. Now Carol can take the following strategy: In the remaining k rounds, she always says that C_2 is the correct chip. Then after the k moves, C_2 is still in the game. But so is C_1, because it takes at least k moves to travel down all the way to the x_0-position and one more to be kicked out. Hence, there are two chips left and Paul cannot decide which one is correct.

Now assume that the weight is $\leq 2^k$. Then C_1 is the only chip, and Paul has already won.

Case 2: $x_k = 0$

First assume that the weight is $\leq 2^k$. Then Paul chooses the question $v := (0, \ldots, 0, \lfloor \frac{1}{2} x_0 \rfloor)$. ($\lfloor \ \rfloor$ means rounding down to the next integer.) The two possible consecutive states differ only at the x_0-position, and it is better for Carole to take $NO(P, v) = (0, x_k, \ldots, x_2, x_1 + \lceil \frac{1}{2} x_0 \rceil)$, having weight

$$
\begin{aligned}
w(NO(P,v)) &= \left\lceil \frac{1}{2} x_0 \right\rceil + x_1 + \sum_{i=1}^{k-1} 2^i x_{i+1} \\
&= \left\lceil \frac{1}{2} x_0 \right\rceil + \sum_{i=1}^{k} 2^{i-1} x_i \\
&= \left\lceil \frac{w(P)}{2} \right\rceil \leq \left\lceil \frac{2^k}{2} \right\rceil = 2^{k-1}.
\end{aligned}
$$

So Paul can assure that in the following state, the weight is $\leq 2^{k-1}$. By induction, after k rounds the weight is ≤ 1, implying that only one chip is left. Hence, Paul wins the game.

Now assume that the weight is $> 2^k$. Paul asks a question, and Carol choses the answer that leaves more chips on the x_0-position. The other positions are indifferent against Carols choice, and the consecutive state is $P_{new} = (0, x_k, \ldots, x_2, x_1 + \tilde{x}_0)$, with some $\tilde{x}_0 \geq \frac{1}{2} x_0$.

Then the weight of the new position is at least

$$
\begin{aligned}
w(P_{new}) &\geq \tilde{x}_0 + x_1 + \sum_{i=1}^{k-1} 2^i x_{i+1} \\
&\geq \frac{1}{2} x_0 + \sum_{i=1}^{k} 2^{i-1} x_i \\
&= \frac{w(P)}{2} > \frac{2^k}{2} = 2^{k-1}.
\end{aligned}
$$

So Carol can assure that in the following state, the weight is $> 2^{k-1}$. By induction, after k rounds the weight is > 1. But during those rounds, all chips must move all the way down to the x_0-position. So all chips have weight 1, implying that there is more than one chip left. Hence, Carol wins the game.

Corollary 8. *If P is a state in the liars game, and if $j \geq k$ with $w_j(P) > 2^j$, then Paul can not win the game within j moves.*

Hence, $\max\{j \geq k | \ w_j(P) \leq 2^j\}$ is a lower bound for the minimal number of questions that Paul needs.

Proof. Assume Paul had a strategy that would yield him victory in j moves. Then Carol does the following: In each round, she picks the answer with the higher weight function.

By the triangle equality, the new weight will be at least half the old weight. Hence, we have the invariant that $w_i(P_i) > 2^i$, where P_i is the state when there are i questions left.

In particular, for $i = k$, we have $w_k(P_k) > 2^k$, and by our assumption, Paul can still win within k moves. This is a contradiction to the starting condition of our theorem.

We now show an almost tight lower bound for the case that $n \leq 2^{2^k}$. To do so, we need the following lemma.

Lemma 9. *For $n = 2$, Paul needs at least $2k + 1$ questions to win the game.*

Proof. For the first k questions Carole claims that $x = 1$, and for the next k questions she claims $x = 2$. Now Paul needs one additional questions to finally determine Carole's choice.

The above lower bound for $n = 2$ extends in the following way to arbitrary n.

Lemma 10. *Paul needs at least $\log n + 2k$ questions to win the game.*

Proof. From the start position $(n, 0, \ldots, 0)$, Paul needs at least $\log n - 1$ questions to reach a position $P = (x_k, \ldots, x_0)$ with $x_k = 2$, if Carole always chooses an answer that yields the largest entry in the first component of the successor position. Lemma 9 implies that Paul needs at least $2k + 1$ questions to win the game from position P.

Thus the total number of questions needed for Paul to win the game is at least $\log n + 2k$.

Theorem 1 (ii) is now a corollary of the lemma above.

Proof (Theorem 1 (ii)). If $n < 2^{q-2k}$ then Paul may ask less than $\log n + 2k$ questions and henceforth cannot win the game by Lemma 10.

References

[BK93] Ryan S. Borgstrom and S. Rao Kosaraju. Comparison-based search in the presence of errors. In *STOC*, pages 130–136, 1993.

[CM99] Ferdinando Cicalese and Daniele Mundici. Optimal binary search with two unreliable tests and minimum adaptiveness. In Jaroslav Nesetril, editor, *ESA*, volume 1643 of *Lecture Notes in Computer Science*, pages 257–266. Springer, 1999.

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

[Pel89] Andrzej Pelc. Searching with known error probability. *Theor. Comput. Sci.*, 63(2):185–202, 1989.

[Pel02] Andrzej Pelc. Searching games with errors - fifty years of coping with liars. *Theor. Comput. Sci.*, 270(1-2):71–109, 2002.

[Rén61] Alfréd Rényi. On a problem of information theory. *MTA Mat. Kut. Int. Kozl.*, 6B:505–516, 1961.

[Rén76] Alfréd Rényi. *Napl'o az információelméletről*. Gondolat, Budapest, 1976. (English translation: A Diary on Information Theory, Wiley, New York, 1984.)

[Spe92] Joel Spencer. Ulam's searching game with a fixed number of lies. *Theor. Comput. Sci.*, 95(2):307–321, 1992.

[Ula76] Stanislaw M. Ulam. *Adventures of a Mathematician*. p. 281, Scribner, New York, 1976.