Two-batch liar games on a general bounded channel

R.B. Ellis\textsuperscript{1}  K.L. Nyman\textsuperscript{2}

\textsuperscript{1}Illinois Institute of Technology
\textsuperscript{2}Loyola University Chicago

BilleraFest
Basic liar game setting

Two-person game:

1. **Carole** picks a number $x \in [n] := \{1, \ldots, n\}$
2. **Paul** asks $q$ questions to determine $x$:
   
   - given $[n] = A_1 \cup A_2 \cup \cdots \cup A_t$,
   - for what $i$ is $x \in A_i$?

Playing optimally, **Carole** answers with an adversarial strategy; it’s a perfect information game.

**Catch:** **Carole** is allowed to lie up to $k$ times.
Example ternary game

\( t = 3 \) (Ternary coding).

- **Paul** partitions \([n] = A_1 \cup A_2 \cup A_3\) and asks “for what \(i\) is \(x \in A_i\)?”
- **Carole** answers 1, 2, or 3

Example. \(n = 6, q = 4, t = 3, k = 1\)

<table>
<thead>
<tr>
<th>Rnd</th>
<th>(A_1)</th>
<th>(A_2)</th>
<th>(A_3)</th>
<th>Carole</th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
<th>(5)</th>
<th>(6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1,2}</td>
<td>{3,4}</td>
<td>{5,6}</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>{3}</td>
<td>{4}</td>
<td>{1,2,5,6}</td>
<td>3</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>{1,2}</td>
<td>{3,4}</td>
<td>{5,6}</td>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>{5}</td>
<td>{6}</td>
<td>\emptyset</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Therefore \(x = 5\).
Binary symmetric case

- $t = 2$ binary case $\iff$ “is $x \in A_1$?”
- symmetric lies: Carole may
  - lie with Yes when truth is No
  - lie with No when truth is Yes

**Question.** Given $q$, what is the maximum $n$ for which Paul has a winning strategy to find $x$?

- $k = 0$, binary search, $n = 2^q$
The basic liar game

Binary symmetric case

- \( t = 2 \) binary case \( \leftrightarrow \) “is \( x \in A_1 \) ?”
- symmetric lies: Carole may
  - lie with Yes when truth is No
  - lie with No when truth is Yes

**Question.** Given \( q \), what is the maximum \( n \) for which Paul has a winning strategy to find \( x \)?

- \( k = 0 \), binary search, \( n = 2^q \)
- \( k = 1 \), Pelc (87); \( k = 2 \), Guzicki (90); \( k = 3 \), Deppe (00)
- \( k < \infty \), Spencer (1992) (*up to bounded additive error*)
**Binary symmetric case**

- $t = 2$ binary case $\leftrightarrow$ “is $x \in A_1$?”
- **Symmetric lies**: Carole may
  - lie with Yes when truth is No
  - lie with No when truth is Yes

**Question.** Given $q$, what is the maximum $n$ for which Paul has a winning strategy to find $x$?

- $k = 0$, binary search, $n = 2^q$
- $k = 1$, Pelc (87); $k = 2$, Guzicki (90); $k = 3$, Deppe (00)
- $k < \infty$, Spencer (1992) (*up to bounded additive error*)
- $k/q \rightarrow f \in (0, 1/2)$, Berlekamp (1962+), Zingangirov
**Binary symmetric case, \( k = 1 \)**

**Question.** Given \( q \), what is the maximum \( n \) for which Paul has a winning strategy to find \( x \)?

- Let \( k = 1 \), Carole chooses \( y \in [n] \)
- \( q + 1 \) possible responses if \( y \) is the distinguished element:

<table>
<thead>
<tr>
<th>0 lies</th>
<th>Game response string ( w \in [2]^q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( w_1 )</td>
<td>( w_2 )</td>
</tr>
<tr>
<td>( \overline{w_1} )</td>
<td>*</td>
</tr>
<tr>
<td>( w_1 )</td>
<td>( \overline{w_2} )</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
</tr>
<tr>
<td>( w_1 )</td>
<td>( w_2 )</td>
</tr>
</tbody>
</table>

**Sphere Bound** \( y, y' \) can’t both be \( x \) \( \implies \) \( n \leq 2^q / \binom{q}{1} \)

where \( \binom{q}{1} = \binom{q}{0} + \binom{q}{1} \).
Binary symmetric case, $k < \infty$

- $(\binom{q}{\leq k})$ response strings corresponding to $y \in [n]$ being the distinguished element

**Sphere Bound**  
$n \leq \frac{2^q}{\binom{q}{\leq k}}$

$X_i :=$ elements of $[n]$ with $i$ accumulated lies

Paul balances $A_1 \cup A_2$ by solving each round

$$|A_1 \cap X_i| = \frac{|X_i|}{2}, \quad \text{for } 0 \leq i \leq k.$$
Asymmetric lying

- **asymmetric lies**: Carole may
  - lie with Yes (1) when truth is No (2)
  - But not vice versa!

Called the \( Z \)-channel

- \( k < \infty \), Dumitriu & Spencer (2004)
- \( k < \infty \) w/improved asymptotics, Spencer & Yan (2003)

Asymmetric strategy: still based on balancing.
A motivating question

(Linial 2005): What if Paul knows that Carole is lying according to one of the $Z$-channels, but not which one?

Our answer: Yes! We generalize the “channel” constraining Carole’s lies as much as possible.
A closer look: game lie strings

<table>
<thead>
<tr>
<th>Rnd</th>
<th>Paul $A_1$</th>
<th>Paul $A_2$</th>
<th>Paul $A_3$</th>
<th>Carole $w$</th>
<th>6’s lie string</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1, 2}</td>
<td>{3, 4}</td>
<td>{5, 6}</td>
<td>2</td>
<td>3, 2</td>
</tr>
<tr>
<td>2</td>
<td>{3}</td>
<td>{4}</td>
<td>{1, 2, 5, 6}</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>{1, 2}</td>
<td>{3, 4}</td>
<td>{5, 6}</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>{5}</td>
<td>{6}</td>
<td>$\emptyset$</td>
<td>1</td>
<td>2, 1</td>
</tr>
</tbody>
</table>

Truthful string for $y = 6$: $w' = 3 \ 3 \ 3 \ 2$

Lie string for $y = 6$: $u = 3 \ 2 \ 2 \ 1$

Game response string: $w = 2 \ 3 \ 3 \ 1$

Write $u = (3, 2)(2, 1)$; we say $w' \xrightarrow{u} w$
The general bounded \( t \)-ary channel

- **Lies**: \( L(t) := \{(a, b) \in [t] \times [t] : a \neq b\} \) (truth = \( a \), Carole: \( b \))
- **Lie strings**: \( L(t)^j := \{(a_1, b_1) \cdots (a_j, b_j) : (a_i, b_i) \in L(t)\} \)
- **Empty string**: \( L(t)^0 := \{\epsilon\} \)

**Definition (General bounded channel)**

Fix \( k \geq 0 \). A channel \( C \) of order \( k \) is an arbitrary subset

\[
C \subseteq \bigcup_{j=0}^{k} L(t)^j,
\]

such that \( C \cap L(t)^k \neq \emptyset \).
Element survival and winning for Paul

Definition
An element $y \in [n]$ survives the game iff its lie string is in $C$.

Definition
Paul wins the original liar game iff at most one element survives after $q$ rounds.
Paul wins the pathological liar game iff at least one element survives after $q$ rounds.

\[ A_C(q) := \max n \quad \text{and} \quad A^*_C(q) := \min n \] such that Paul can win the original pathological liar game with $n$ elements.
Example channels

- **Binary, symmetric, two lies.** \((t = 2, k = 2)\)
  \[ C = \{ \epsilon, (1, 2), (2, 1), (1, 2)(1, 2), (1, 2)(2, 1), (2, 1)(2, 1), (2, 1)(1, 2) \} \]
  \[ \frac{2^q}{\binom{q}{\leq 2}} - O(1) = A_C(q) \leq A^*_C(q) = \frac{2^q}{\binom{q}{\leq 2}} + O(1) \]
  Guzicki ('90); Ellis, Ponomarenko, Yan ('05)

- **Binary, Z-channel, two lies.** \((t = 2, k = 2)\)
  \[ C = \{ \epsilon, (2, 1), (2, 1)(2, 1) \} \]
  \[ A_C(q), A^*_C(q) \sim \frac{2^{q+2}}{\binom{q}{\leq 2}}, \quad \text{Spencer, Yan ('03); here} \]
Example channels (con’t)

- Binary, unidirectional, two lies. \((t = 2, k = 2)\)

\[
C = \{\epsilon, (1, 2), (2, 1), (1, 2)(1, 2), (2, 1)(2, 1)\}
\]

\[
A_C(q), A^*_C(q) \sim \frac{2^{q+1}}{\binom{q}{\leq 2}}, \text{ here}
\]

- Selective lies.
  - Pick arbitrary \(L' \subseteq L(t)\).
  - Let \(C = \bigcup_{j=0}^{k} (L')^j\).

\[
A_C(q), A^*_C(q) \sim \frac{t^{q+k}}{|L'|^k \binom{q}{\leq k}}, \text{ here}
\]

Dumitriu, Spencer (‘05); here
Example (weighted lies).
- Weight the lies of $L(t)$, normalized to minimum weight 1.
- Let $k$ bound the total allowable weight of a game lie string.
- Let $C = \{ u \in L(t) \geq 0 : \text{weight}(u) \leq k \}$.

$A_{C,t}(q)$ was solved asymptotically by Alshwede, Cicalese, & Deppe (2006+); slightly improved here.

Example (Model-based channel).
- Select a communication model (probability map $p : L(t) \geq 0 \rightarrow [0, 1]$).
- Select a probability threshold $p_0$.
- Let $C = \{ u \in L(t) \geq 0 : p(u) > p_0 \}$.

Paul must handle all likely errors/lie strings.
The proposed sphere bound

- **Select** Paul’s strategy tree to be *random partitions* so the truthful response string is random.

- Carole picks a lie string \( u \in C \), and places to put the lies.

\[
\begin{array}{l}
\text{Truthful string for } y & w' = w'_1 \cdots w'_{i_1} \cdots w'_{i_\ell} \cdots w'_{i_j} \cdots w'_q \\
\text{Lie string for } y & u = a_1 \quad a_\ell \quad a_j \\
\text{Response string} & w = w_1 \cdots w_{i_1} \cdots w_{i_\ell} \cdots w_{i_j} \cdots w_q \\
\end{array}
\]

- **Compatibility:** \( \Pr(w'_{i_\ell} = a_\ell) = t^{-1} \)
The proposed sphere bound

- The expected number of response strings for which $y$ survives is:

$$\sum_{u \in C} \binom{q}{|u|} t^{-|u|} \sim |C \cap L(t)^k| \binom{q}{k} t^{-k}.$$

**Definition (Asymptotic Sphere Bound)**

For $q$ rounds, base $t$, and an order $k$ channel $C$, the sphere bound is

$$SB_C(q) := \frac{t^{q+k}}{|C \cap L(t)^k| \binom{q}{k}}.$$
Carole’s bound

Theorem (Carole’s bound)

\[ A_C(q) \leq \text{SB}_C(q)(1 + o(1)), \]
\[ A_C^*(q) \geq \text{SB}_C(q)(1 - o(1)). \]

Proof idea.
- Get lower and upper bounds on the number of response strings for which an element \( y \) survives.
- If \( n \) is too large, the response string sets collide. If Carole responds with a string in the intersection, Paul cannot be sure which element Carole was thinking of.
- If \( n \) is too small, the response strings fail to cover \([t]^q\).
Paul’s bound

Theorem (Paul’s bound)

\[ A_C(q) \geq SB_C(q)(1 - o(1)), \]
\[ A^*_C(q) \leq SB_C(q)(1 + o(1)). \]

Furthermore, we may restrict Paul to two nonadaptive batches of questions of sizes \( q_1 \) and \( q_2 \), with

\[ q_1 + q_2 = q \quad \text{and} \quad (\log_t q)^{3/2} << q_2 \leq \text{cst} \cdot q^{k/(2k-1)}, \]

Remark. Proof builds on techniques of Dumitriu&Spencer.
(\(M, r\))-balanced strings in \([t]^Q\)

\[
\frac{1}{t} \left\lfloor \frac{Q}{M} \right\rfloor - r(t-1) - \Theta(1) \leq \#1's, \#2's \leq \frac{1}{t} \left\lfloor \frac{Q}{M} \right\rfloor + r
\]

- By counting the number of ways to place lies in sections we can bound \(|\{w' : w' \xrightarrow{u} w\}|\).

**Lemma**

Let \(u = (a_1, b_1) \cdots (a_j, b_j)\), and \(w \in [t]^Q\) be \((M, r)\)-balanced. Then

\[
\binom{M}{j} \left( \frac{1}{t} \left\lfloor \frac{Q}{M} \right\rfloor - r(t-1) - \Theta(1) \right)^j \leq |\{w' : w' \xrightarrow{u} w\}| \leq \binom{M+j-1}{j} \left( \frac{1}{t} \left\lfloor \frac{Q}{M} \right\rfloor + r \right)^j
\]
First batch of $q_1$ questions

(Proof illustrated with $C = \{\epsilon, (1, 2), (2, 1), (1, 2)(1, 2), (2, 1)(2, 1)\}$.)

Paul maps $n$ evenly to $(M, r)$-balanced vertices of $[t]^{q_1}$

Paul asks: What is the $i^{th}$ coordinate in your element’s length-$q_1$ string?

$$n = SB_C(q)(1 - o(1))$$
Carole’s first batch response

Suppose Carole responds with balanced \( w \in [t]^{q_1} \).
Which \( y \in [n] \) survive?

Any \( y \) identified with \( w' \) such that:
- \( u \in C \), and
- \( w' \xrightarrow{u} w \)
Paul’s second batch of $q_2$ questions

- $y$’s survive in various ways
- Fit $y$’s which can take more lies inside disjoint Hamming balls
- $(M, r)$-balance $\Rightarrow$ control on $|\{w^{(i)} : w^{(i)} \rightarrow w\}|$, $|\{z : z \rightarrow z'\}|$
- Greedily pack other $y$’s in unoccupied space
First batch, pathological case

(Proof illustrated with $C = \{\epsilon, (1, 2), (2, 1), (1, 2)(1, 2), (2, 1)(2, 1)\}$.)
First batch, pathological case

(Proof illustrated with $C = \{\epsilon, (1, 2), (2, 1), (1, 2) (1, 2), (2, 1) (2, 1)\}$.)

- Paul adds negligibly many elements evenly over $[t]^{q_1}$
Paul’s second batch, pathological case

\[ \begin{array}{c}
\begin{array}{c}
\varepsilon \\
(1,2)
\end{array} \\
W^{(1)} \\
(1,2)(1,2)
\end{array}
\begin{array}{c}
\varepsilon \\
W^{(2)} \\
(2,1)(2,1)
\end{array}
\begin{array}{c}
\varepsilon \\
W^{(3)} \\
(2,1)(2,1)
\end{array}
\begin{array}{c}
\varepsilon \\
W^{(4)} \\
(2,1)
\end{array}
\end{array} \]
Paul’s second batch, pathological case

- Count only additional \( y \)'s for which Carole may not lie again
- Greedily convert packing into covering in \( [t]^{q_2} \)
Summary

**Theorem**

\[ SB_C(q)(1 + o(1)) \geq A_C(q) \geq SB_C(q)(1 - o(1)), \]
\[ SB_C(q)(1 - o(1)) \leq A^*(q) \leq SB_C(q)(1 + o(1)). \]

Furthermore, (1) we may restrict Paul to two nonadaptive batches of questions of sizes \( q_1 \) and \( q_2 \), with

\[ q_1 + q_2 = q \quad \text{and} \quad (\log q)^{3/2} \ll q_2 \leq \text{cst} \cdot q^{k/(2^k - 1)}, \]

(2) the response sets for \( A_C(q) \) are a subset of those for \( A^*_C(q) \).
Open Questions.

- Can we further **reduce** or **eliminate** completely the adaptiveness?
- Can these techniques be used to improved the **asymptotic best known packings and coverings** of $[t]^q$ with fixed-radius Hamming balls (not tight for radius $\geq 2$)?
- Will these techniques work for **coin-weighing**, **fault-testing**, and related **search problems**?
Happy Birthday, Lou!
....and thank you!