Optimal Short-Circuit Resilient Formulas

Ran Gelles
Bar-Ilan University

Mark Braverman
Princeton University

Klim Efremenko
Ben-Gurion Univ.

Michael A. Yitayew
Princeton University
Motivation

• How to construct a circuit that computes

\[ f(z) = z_1 \land z_2 \]

• Assuming AND / OR gates (all negations pushed to literals)

• EASY:
Motivation
Motivation
Motivation

• How to construct a circuit that computes

\[ f(z) = z_1 \land z_2 \]

• Assuming AND / OR gates

• When few of the AND / OR gates were mixed?
Motivation

• A More General Question:
  
  • given a boolean function
    \[ f(z) : \{0, 1\}^n \rightarrow \{0, 1\} \]
    
  • construct an AND / OR circuit for \( f \), that works
    **even if a constant fraction of the gates are “faulty”**
Short-Circuit Noise

- A generalization of the above is a faulty gate with “short-circuit” noise
- The shorted input can be determined adversarially
- Equivalent to replacing the gate with an arbitrary gate $g$ for which $g(0...0) = 0$ and $g(1...1) = 1$
Short-Circuit Noise

• This type of noise is very common in produced wafers

• Incomparable to von-Neumann Noise (every wire flips w.p $\epsilon$)
Short-Circuit Noise

• Main question(s):
  • How to construct an AND/OR circuit that is correct with up to \( k \) faulty (short-circuited) gates
  • What is the maximal \( k \)?
  What is the maximal fraction of faulty gates?
  • How many extra gates we need to “fortify” a given circuit?
## Prior Work

<table>
<thead>
<tr>
<th>Work</th>
<th>Noise level</th>
<th>Circuit</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kleitman-Leighton-Ma</td>
<td>$k$ errors</td>
<td>any</td>
<td>$O(k ,</td>
</tr>
<tr>
<td>(J.Comp.Sys.Sci97)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kalai-Lewko-Rao</td>
<td>$\delta &lt; 1/6$ fraction</td>
<td>formula (fan-in$&gt;$2)</td>
<td>poly($</td>
</tr>
<tr>
<td>(FOCS12)</td>
<td>$\delta &lt; 1/10$ fraction</td>
<td>formula (fan-in=2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(*in-to-out path)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Resilient Circuits with Von Neuman Noise:**
Resilient Formulas

• The Attack Plan: [Kalai-Lewko-Rao 2012]
Resilient Formulas

• Why do we lose a factor-2 in the resilience?

• Noise is one-sided:
  • Noise on **AND** gates can only make **0→1**
  • Noise on **OR** gates can only make **1→0**

• If out=1, noise on **AND** gates is meaningless!

• If a circuit is resilient to **δ’**-fraction, then
  (1) corrupting **δ’**-fraction of **ANDs** is OK, but also
  (2) corrupting **δ’**-fraction of **ORs** is OK

⇒ is resilient to **2δ’**, thus **2δ’ ≤ δ** (since res. comes from protocol)
Resilient Formulas

• Idea: split the noise to AND and OR gates

• Def. \((\alpha, \beta)\)-corruption means corrupting at most \(\alpha n\) AND gates and \(\beta n\) OR gates in every in-to-out path (\(n\) is depth of circuit)
Result

\[(\alpha, \beta)\)-resilience

\[(1/5, 1/5)\)-resilient coding + converse

[KW90]

[EGH16]

[KLR12]
Main Result

- **Upper Bound (Direct):**
  Any formula $F$ can be (efficiently) compiled into $F'$ so that:
  - $F'$ is correct if up to $\frac{1}{5} - \epsilon$ fraction of the AND-gates, and $\frac{1}{5} - \epsilon$ of the OR-gates are faulty in any input-to-output branch
  - $F'$ has constant fan in ($> 2$), $|F'| = \text{poly}(|F|)$

- **Lower Bound (Converse):** Resilience of $1/5$ is **tight**.
  There exist functions that $1/5$ corruption invalidates any $F$ of sub-exponential size
Techniques: Upper Bound

(1/5, 1/5)-resilient coding scheme w/ feedback
Coding for Interactive Comm.

\[ \pi(x, y) \]

Many Coding Schemes exist for various settings

[Schulman96, GMS14, BR14, KR13, GH15, Pan13, EGH16, Hau14, BK12, BKN14, FGOS15, BGO16, BNTTU14, G17, GHKZW18] ...
Feedback

• We define a noisy *KW mapping* between formulas and protocols

• Short Circuit noise == Channel noise
  (assuming feedback)

• The *sender* learns the received symbol via a “noiseless feedback“ channel
Coding Scheme - Overview

• Assume a noiseless binary protocol $\pi$

• Alice and Bob simulate $\pi$ message by message. Each message contains:
  • the “next” bit according to $\pi$
  • a link to the previous non-corrupt message sent by the party (as learnt by feedback)

• Each received message induces a “chain” of allegedly correct messages. The next step follows this chain

• At the end, the longest chain is to be trusted
Coding Scheme

Aim: simulate the noiseless protocol step-by-step

Alice doesn’t know there was an error. gives wrong info
Bob knows this is wrong (via feedback)

This extension ignores $m_5$,

Bob “knows” $m_5$ is based on err

Bob knows this is wrong (via feedback)
Coding Scheme

Output: the transcript implied by the longest chain

Alice received $m_6$. Based on it she “knows” $m_4$ is an err, and she knows $m_5$ is to be ignored.
Coding Scheme

- Messages are **not** alternating order
- the **more** noise on Bob’s messages, the **less** he gets to speak in the future
Attacks (1)

• Adversary may try to build its own chain

• But with $1/5$-fraction corruptions, his chain will be shorter
Attacks (2)

- Adversary may incorrectly extend a correct chain.
- But in order to make its chain the longest, it must start late.
- By then, the chain’s already simulated the entire transcript.
Techniques: Lower Bound
Lower Bound

• Note, $(1/5, 1/5)$-corruptions cannot fool protocols with exponential (blowup in) communication:

  • Use Shannon code with relative distance $\approx 1$ to exchange the parties inputs.

  • Withstands noise rate of $\approx 1/2$ per direction of the channel
Lower Bound

• Yet, when the blow-up is restricted (e.g., communication < size of the input):

• By a Pigeonhole principle, we can show a function $f$ and inputs $x,y,x',y'$ for which

1. $f(x,y) \neq f(x',y) \neq f(x',y')$

• If the computation of $f$ takes $r$ rounds by some protocol, then during its first $2r/5$ rounds:

2. Alice (wlog) speaks at most half of the times

3. If Alice has $x$, then the protocol sends exactly the same messages whether Bob holds $y$ or $y'$
Lower Bound

• Create the following confusing transcript:

\[ \pi(x,y) \quad \pi(x',y) \quad \pi(x',y') \quad \pi(x',y) \]

- Bob speaks \[ \leq r/5 \]
- Alice speaks \[ \leq r/5 \]
- Rounds \[ 2r/5 \]
- \[ r/5 \quad r/5 \]

(until terminates, if hasn’t already)

• is a \((1/5,1/5)\)-corruption of \(\pi(x',y)\) and one of \(\{\pi(x',y'), \pi(x,y)\}\)
Lower Bound

- Example: Assume $\pi$ terminates before 4-th part

\[
\begin{align*}
(x', y) & \quad \pi(x, y) \quad \pi(x', y) \quad \pi(x', y') \\
(x', y') & \quad \pi(x, y) \quad \pi(x', y) \quad \pi(x', y')
\end{align*}
\]

\[\pi(x, y) = \pi(x, y')\] from (3) of pigeon hole

Since $f(x', y) \neq f(x', y')$ we are done (1) of pigeon hole...
Lower Bound

• Problem:
  • Need to apply the above on \textit{KW-relation}, rather than on a \textit{function}.
  • \( f(x',y) \neq f(x',y') \) translates to confusing Alice between \( KW(x',y) \) and \( KW(x',y') \)
  • but maybe \textbf{both} are a correct output of the protocol?!
  • We use KW relation of the \textit{parity function} \( par(x_1,\ldots,x_n) = x_1 \oplus \cdots \oplus x_n \), choosing inputs so that
    
    \[
    KW_{\text{par}}(x',y) \cap KW_{\text{par}}(x',y') = \emptyset
    \]

Summary
Summary

• A two-directional “noisy” KW mapping between protocols and formulas

• Coding scheme with resilience $1/5 - \varepsilon$ (const alphabet)
  ➔ Formula resilient to $(1/5 - \varepsilon, 1/5 - \varepsilon)$–noise

• Impossibility of coding with $1/5$ (const rate)
  ➔ No small formula is resilient to $(1/5, 1/5)$-noise
Open Problems

1. The binary / fan-in2 case?

2. General faults: stuck to 0/1, flip, short-circuit

3. KW connects *formulas* with *2-party* protocols
   - Can we map general *circuits* with some kind of communication model?
   - (Branching Programs? multiparty protocols?)
The End...