Bounds on Reliable Boolean Function Computation with Noisy Gates

- R. L. Dobrushin & S. I. Ortyukov, 1977
- N. Pippenger, 1985
- P. Gács & A. Gál, 1994

Presenter: Da Wang
6.454 Graduate Seminar in Area I
EECS, MIT
Oct. 5, 2011
Question

*Given a network of noisy logic gates, what is the redundancy required if we want to compute the a Boolean function reliably?*

- **noisy**: gates produce the wrong output independently with error probability no more than $\varepsilon$.
- **reliably**: the value computed by the entire circuit is correct with probability at least $1 - \delta$.
- **redundancy**: minimum #gates needed for reliable computation in noisy circuit
  
  minimum #gates needed for reliable computation in noiseless circuit

- noisy/noiseless complexity
- may depend on the function of interest
- upper bound: achievability
- lower bound: converse
Part I

Lower Bounds for the Complexity of Reliable Boolean Circuits with Noisy Gates
History of development

- [Dobrushin & Ortyukov 1977]
  - Contains all the key ideas
  - Proofs for a few lemmas are incorrect
- [Pippenger & Stamoulis & Tsitsiklis 1990]
  - Pointed out the errors in [DO1977]
  - Provide proofs for the case of computing the parity function
- [Gács & Gál 1994]
  - Follow the ideas in [DO1977] and provide correct proofs
  - Also prove some stronger results

In this talk
We will mainly follow the presentation in [Gács & Gál 1994].
Problem formulation

System Model

**Boolean circuit** $C$
- a directed acyclic graph
- node $\sim$ gate
- edge $\sim$ in/out of a gate

**Gate** $g$
- a function $g : \{0, 1\}^{n_g} \rightarrow \{0, 1\}$
  - $n_g$: fan-in of the gate

**Basis** $\Phi$
- a set of possible gate functions
- e.g., $\Phi = \{AND, OR, XOR\}$
- complete basis
- for circuit $C$: $\Phi_C$
- maximum fan-in in $C$: $n(\Phi_C)$

**Assumptions**
- each gate $g$ has constant number of fan-ins $n_g$.
- $f$ can be represented by compositions of gate functions in $\Phi_C$. 
Problem formulation

Error models $(\varepsilon, p)$

**Gate error**
- A gate fails if its output value for $z \in \{0, 1\}^{n_g}$ is different from $g(z)$
- Gates fail independently with
  - fixed probability $\varepsilon$
    - used for lower bound proof
  - probability at most $\varepsilon$
- $\varepsilon \in (0, 1/2)$

**Circuit error**
- $C(x)$: random variable for output of circuit $C$ on input $x$.
- A circuit computes $f$ with error probability at most $p$ if
  \[ \mathbb{P}[C(x) \neq f(x)] \leq p \]
  for any input $x$. 
Problem formulation

Sensitivity of a Boolean function

Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be a Boolean function with binary input vector $x = (x_1, x_2, \ldots, x_n)$.
Let $x^l$ be a binary vector that differs from $x$ only in the $l$-th bit, i.e.,

$$x^l_i = \begin{cases} 
x_i & i \neq l \\
-x_i & i = l.
\end{cases}$$

- $f$ is sensitive to the $l$th bit on $x$ if $f(x^l) \neq f(x)$.
- Sensitivity of $f$ on $x$: #bits in $x$ that $f$ is sensitive to.
  - “effective” input size
- Sensitivity of $f$: maximum over all $x$. 
Asymptotic notations

- \( f(n) = O(g(n)) \):
  \[ \lim_{n \to \infty} \left| \frac{f(n)}{g(n)} \right| < \infty, \]

- \( f(n) = \Omega(g(n)) \):
  \[ \lim_{n \to \infty} \left| \frac{f(n)}{g(n)} \right| \geq 1, \]

- \( f(n) = \Theta(g(n)) \):
  \[ f(n) = O(g(n)) \]
  and
  \[ f(n) = \Omega(g(n)) \]
Main results

**Theorem: number of gates for reliable computation**

- Let $\varepsilon$ and $p$ be any constants such that $\varepsilon \in (0, 1/2), p \in (0, 1/2)$.
- Let $f$ be any Boolean function with sensitivity $s$.

Under the error model $(\varepsilon, p)$, the number of gates of the circuit is $\Omega(s \log s)$.

**Corollary: redundancy of noisy computation**

For any Boolean function of $n$ variables and with $O(n)$ noiseless complexity and $\Omega(n)$ sensitivity, the redundancy of noisy computation is $\Omega(\log n)$.

- e.g., nonconstant symmetric function of $n$ variables has redundancy $\Omega(\log n)$
Equivalence result for wire failures

**Lemma 3.1 in Dobrushin&Ortyukov**

- Let $\varepsilon \in (0, 1/2)$ and $\delta \in [0, \varepsilon/n(\Phi_C)]$.
- Let $y$ and $t$ be the vector that a gate receives when the wire fail and does not fail respectively.

For any gate $g$ in the circuit $C$ there exists unique values $\eta_g(y, \delta)$ such that if
- the wires of $C$ fails independently with error probability $\delta$, and
- the gate $g$ fails with probability $\eta_g(y, \delta)$ when receiving input $y$,
then the probability that the output of $g$ is different from $g(t)$ is equal to $\varepsilon$.

**Insights**

- Independent gate failures can be “simulated” by independently wire failures and corresponding gate failures.
- These two failure modes are equivalent in the sense that the circuit $C$ computes $f$ with the same error probability.
Theorem

- Let $\varepsilon$ and $p$ be any constants such that $\varepsilon \in (0, 1/2), p \in (0, 1/2)$.
- Let $f$ be any Boolean function with sensitivity $s$.

Let $C$ be a circuit such that

- its wires fail independently with fixed probability $\delta$, and
- each gate fails independently with probability $\eta_g(y, \delta)$ when receiving $y$.

Suppose $C$ computes $f$ with error probability at most $p$. Then the number of gates of the circuit is $\Omega(s \log s)$. 
Error analysis
Function and circuit inputs

**Maximal sensitive set $S$ for $f$**

- $s > 0$: sensitivity of $f$
- $z$: an input vector with $s$ bits that $f$ is sensitive to
  - an input vector where $f$ has maximum sensitivity
- $S$: the set of sensitive bits in $z$
  - key object

**$B_l$: edges originated from $l$-th input**

- $m_l \triangleq |B_l|$
- e.g.
  - $l = 3$
  - $B_l$
  - $m_l = 3$
Error analysis

Wire failures

- For $\beta \subset B_l$, let $H(\beta)$ be the event that for wires in $B_l$, only those in $\beta$ fail.

- Let

$$\beta_l \triangleq \arg \max_{\beta \subset B_l} \mathbb{P} \left[ C(z^l) = f(z^l) \mid H(\beta) \right]$$

  ▶ the best failing set for input $z^l$

- Let $H_l \triangleq H(B_l \setminus \beta_l)$

**Fact 1**

$$\mathbb{P} \left[ C(z) \neq f(z) \mid H_l \right] = \mathbb{P} \left[ C(z^l) = f(z^l) \mid H(\beta_l) \right]$$

- Proof
  
  ▶ $f$ is sensitive to $z_l$
  
  ▶ $\neg z_l \iff$ “flip” all wires in $B_l$

- $\beta_l$ is the worst non-failing set for input $z$
Error analysis
Error probability given wire failures

**Fact 2**

\[
\mathbb{P}[C(z^l) = f(z^l) | H(\beta_l)] \geq 1 - p
\]

Proof
- \(\mathbb{P}[C(z^l) = f(z^l)] \geq 1 - p\)
- \(\beta_l\) maximizes \(\mathbb{P}[C(z^l) = f(z^l) | H(\beta)]\)

**Fact 1 & 2 \Rightarrow Fact 3**

For each \(l \in S\),

\[
\mathbb{P}[C(z) \neq f(z) | H_l] \geq 1 - p
\]

where \(\{H_l, l \in S\}\) are independent events. Furthermore, Lemma 4.3 in [Gács&Gál 1994] shows

\[
\mathbb{P}\left[ C(z) \neq f(z) \left\| \bigcup_{l \in S} H_l \right\| \right] \geq (1 - \sqrt{p})^2
\]

The error probability given \(H_l\) or \(\bigcup_{l \in S} H_l\) is relatively large.
Error analysis

Bounds on wire failure probabilities

Note

\[ p \geq \mathbb{P}[C(z) \neq f(z)] \]
\[ \geq \mathbb{P}\left[C(z) \neq f(z) \bigg| \bigcup_{l \in S} H_l\right] \mathbb{P}\left[\bigcup_{l \in S} H_l\right] \]

Fact 3 implies

Fact 4

\[ \mathbb{P}\left[\bigcup_{l \in S} H_l\right] \leq \frac{p}{(1 - \sqrt{p})^2} \]

which implies (via Lemma 4.1 in [Gács&Gál 1994]),

Fact 5

\[ \mathbb{P}\left[\bigcup_{l \in S} H_l\right] \geq \left(1 - \frac{p}{(1 - \sqrt{p})^2}\right) \sum_{l \in S} \mathbb{P}[H_l] \]
Error analysis

Bounds on the total number of sensitive wires

**Fact 6**

\[ \mathbb{P}[H_l] = (1 - \delta)\beta_l \delta^m - |\beta_l| \geq \delta^m_l \]

**Fact 4 & 5 ⇒

\[
\frac{p}{1 - 2\sqrt{p}} \geq \sum_{l \in S} \delta^m_l \\
\geq s \left( \prod_{l \in S} \delta^m_l \right)^{1/s}
\]

which leads to

\[
\sum_{l \in S} m_l \geq \frac{s}{\log(1/\delta)} \log \left( s \frac{1 - 2\sqrt{p}}{p} \right)
\]

lower bound on the total number of “sensitive wires”
Let $N_C$ be the total number of gates in $C$:

\[
n(\Phi_C) N_C \geq \sum_{g} n_g \geq \sum_{l \in S} m_l \geq \frac{s}{\log(1/\delta)} \log \left( s \frac{1 - 2\sqrt{p}}{p} \right)
\]

**Comments:**
- The above proof is for $p \in (0, 1/4)$
- The case $p \in (1/4, 1/2)$ can be shown similarly.
Let \( x^S \) be a binary vector that differs from \( x \) in the \( S \) subset of indices, i.e.,

\[
x^S_i = \begin{cases} 
  x_i & i \not\in S \\
  -x_i & i \in S 
\end{cases}
\]

- \( f \) is (block) sensitive to \( S \) on \( x \) if \( f(x^S) \neq f(x) \).
- **Block sensitivity** of \( f \) on \( x \): the largest number \( b \) such that
  - there exists \( b \) disjoint sets \( S_1, S_2, \ldots, S_b \)
  - for all \( 1 \leq i \leq b \), \( f \) is sensitive to \( S_i \) on \( x \)
- **Block sensitivity** of \( f \): maximum over all \( x \).
  - block sensitivity \( \geq \) sensitivity

**Theorem based on block sensitivity**

- Let \( \varepsilon \) and \( p \) be any constants such that \( \varepsilon \in (0, 1/2) \), \( p \in (0, 1/2) \).
- Let \( f \) be any Boolean function with block sensitivity \( b \).

Under the error model \((\varepsilon, p)\), the number of gates of the circuit is \( \Omega(b \log b) \).
Discussions
Lower bound for specific functions

Given an explicit function $f$ of $n$ variables, is there a lower bound that is stronger than $\Omega(n \log n)$?

Open problem for

- unrestricted circuit $C$ with complete basis
- function $f$ that have $\Omega(n \log n)$ noiseless complexity for circuit $C$ with some incomplete basis $\Phi$
Discussions

Computation model

**Exponential blowup**
A noisy circuit with multiple levels

- The output of gates at level $l$ goes to a gate at level $l + 1$
- Level 0 has $n$ inputs
  - Level 0 has $N_0 = n \log n$ output gates
  - Level 1 has $N_0$ inputs
  - Level 1 has $N_1 = N_0 \log N_0$ output gates, ...

**Why?**
“The theorem is generally applicable only to the very first step of such a fault tolerant computation”

- If the input is not the original ones, we can choose them to make the sensitivity of a Boolean function to be 0.
  - $f(x_1, x_2, x_3, x_4, x_1 \oplus x_2 \oplus x_4, x_1 \oplus x_3 \oplus x_4, x_2 \oplus x_3 \oplus x_4)$
  - Lower bound does not apply: sensitivity is 0. How about block sensitivity?
- Problem formulation issue on the lower bound for **coded** input
  - coding is also computation!
Part II

Upper Bounds for the Complexity of Reliable Boolean Circuits with Noisy Gates

Overview

Achievability schemes in reliable computation with a network of noisy gates.

1. System modeling
   ▶ various types of computations
2. Change of basis and error levels
   ▶ will skip
3. Functions with logarithmic redundancy
   ▶ with explicit construction
   ▶ for specific system parameters only
4. Functions with bounded redundancy
   ▶ Presents a class of functions with “bounded redundancy”
   ▶ Construction for reliable computation
System model: a revisit
Weak vs. strong computation

perturbation and approximation
Let \( f, g : \{0, 1\}^k \rightarrow \{0, 1\} \),

- \( g \) is a \( \varepsilon \)-perturbation of \( f \) if \( \Pr[\text{\( g(x) = f(x) \)}] = 1 - \varepsilon \) for any \( x \in \{0, 1\}^k \)
- \( g \) is a \( \varepsilon \)-approximation of \( f \) if \( \Pr[\text{\( g(x) = f(x) \)}] \geq 1 - \varepsilon \) for any \( x \in \{0, 1\}^k \)

weakly \((\varepsilon, \delta)\)-computes
- gates: \( \varepsilon \)-perturbation
- output: \( \delta \)-approximation

strongly \((\varepsilon, \delta)\)-computes
- gates: \( \varepsilon \)-approximation
- output: \( \delta \)-approximation

Why bother?
- \( \varepsilon \)-perturbation may be helpful in randomized algorithms.
Functions with logarithmic redundancy

Main theorem

**Theorem 3.1**
If a Boolean function is computed by a noiseless network of size $c$, then it is also computed by a noisy network of size $O(c \log c)$.

**Comments**
- Provides explicit construction for some $\varepsilon$ and $\delta$ values.
  - $\varepsilon = 1/512$
  - $\delta = 1/128$
Strategy
Given a noiseless network with 2-input gates, construct a corresponding noisy network with 3-input gates.

Transformations

<table>
<thead>
<tr>
<th>noiseless</th>
<th>noisy</th>
</tr>
</thead>
<tbody>
<tr>
<td>each wire → cable of ( m ) wires</td>
<td></td>
</tr>
<tr>
<td>gate → module of ( O(m) ) noisy gates</td>
<td></td>
</tr>
</tbody>
</table>

Additions

- **coda**: computes the majority of \( m \) wires with at most some error probability
  - Corollary 2.6: exists coda with size \( O(c \log c) \)
- Choose \( m = O(\log c) \)
- a cable is correct if at least \( (1 - \theta)m \) component wires are correct
Module requirement
If the input cables are “correct”, then the output cable will be correct except for some small error probability.
Module construction

Executive organ
- Construction: $m$ noisy gates that compute the same function as the corresponding gate in noiseless network

Restoring organ
- Construction: a $(m, k, \alpha, \beta)$-compressor
  - if at most $\alpha m$ inputs are incorrect, then at most $\beta m$ outputs will be incorrect.
- $k = 8^{17}$, $\alpha = 1/64$, $\beta = 1/512$

Then
Choose system parameters properly, such that the resulting circuit has logarithmic redundancy.
Functions with bounded redundancy

Main results

Functions with bounded redundancy
For $r \geq 1$, let $s = 2^r$. Let

$$g_r(x_0, \ldots, x_{r-1}, y_0, \ldots, y_{s-1}) = y_t$$

where $t = \sum_{i=0}^{r-1} 2^i x_i$ i.e., $t$ has binary representation $x_{r-1} \cdots x_1 x_0$.

Theorem 4.1
For every $r$ and $s = 2^r$, $g_r$ can be computed by a network of $O(s)$ nosiy gates.

Comments
- $g_r$: “indicator function”
- Any noiseless networks that computes $g_r$ has $\Omega(2^r)$ gates.
  - bounded redundancy
- Proof
  - Construct a network that strongly $(\varepsilon = 1/192, \delta = 1/24)$-computes $g_r$. 
Construction

\[ g_1 \]

\[ g_1(x_0, y_0, y_1) = \begin{cases} 
    y_0 & x_0 = 0 \\
    y_1 & x_1 = 1 
\end{cases} \]

\[ g_r \]

\[ g_2(x_0, x_1, y_0, y_1, y_2, y_3) = \begin{cases} 
    y_0 & x_1x_0 = 00 \\
    y_1 & x_1x_0 = 01 \\
    y_2 & x_1x_0 = 10 \\
    y_3 & x_1x_0 = 11 
\end{cases} \]

\[ \ldots \]

\( g_r \) can be implemented by a binary tree with \( 2^r - 1 \) elements of \( g_1 \).

- level \( r - 2 \): root
- level 0: leaves
- \( y_t \): corresponds to a path from level 0 to \( r - 2 \)
Each path only contains one gate at each level
If each gate at level \( k \), \( 0 \leq k \leq r - 2 \) fails with probability \( \Theta \left( (a\varepsilon)^k \right) \), then
the failure probability for a path is \( \Theta \left( \varepsilon \right) \).

Construction: replace wires by cables, gates by modules
- **cable** at level \( k \)
  - input: \( 2k - 1 \) wires
  - output: \( 2k + 1 \) wires
- **module** at level \( k \)
  - \( 2k + 1 \) disjoint networks
  - each compute the \((2k - 1)\)-argument majority of the input wires
  - then apply \( g_1 \)
  - noiseless complexity: \( O(k) \) ⇒ noisy complexity: \( O(k \log k) \)
    - \( O \left( k^2 \log k \right) \) noisy gates at level \( k \)
  - error probability for each nosiy network: \( 2\varepsilon \)
    - error probability for module: \( 4\varepsilon (8\varepsilon)^k = \Theta \left( (8\varepsilon)^k \right) \)
- use **coda** at the root output for majority vote
- total #gate: \( O(s) = O \left( 2^r \right) \)
A network with outputs \(w_1, w_2, \ldots, w_m\) strongly \((\varepsilon, \delta)\)-computes \(f_1, f_2, \ldots, f_m\) if, for every \(1 \leq j \leq m\), the network obtained by ignoring all but the output \(w_j\) strongly \((\varepsilon, \delta)\)-computes \(f_j\).

**Theorem 4.2**

For every \(a \geq 1\) and \(b = 2^{2^a}\), let \(h_{a,0}(z_0, \ldots, z_{a-1}), \ldots, h_{a,b-1}(z_0, \ldots, z_{a-1})\) denote the \(b\) Boolean functions of \(a\) Boolean argument.

Then \(h_{a,0}(z_0, \ldots, z_{a-1}), \ldots, h_{a,b-1}(z_0, \ldots, z_{a-1})\) can be strongly computed by a network of \(O(b)\) noisy gates.

Proof: similar to Theorem 4.1
Boolean function with \( n \) Boolean arguments

**Theorem 4.3**

Any Boolean function of \( n \) Boolean arguments can be computed by a network of \( O\left(\frac{2^n}{n}\right) \) noisy gates.

**Proof**

- Let \( a = \lfloor \log_2(n - \log_2 n) \rfloor \), \( b = 2^{2^a} = 2^n/n \), \( r = n - a \) and \( s = 2^r = 2^n/n \).

- **Theorem 4.2:** \( M \) strongly computes \( h_{a,0}(z_0, \cdots, z_{a-1}), \cdots, h_{a,b-1}(z_0, \cdots, z_{a-1}) \)
  
  
  \( O(b) = O\left(\frac{2^n}{n}\right) \) gates

- **Theorem 4.1:** \( N \) strongly computes
  
  \( g_r(x_0, \cdots, x_{r-1}, y_0, \cdots, y_{s-1}) \)
  
  \( O(s) = O\left(\frac{2^n}{n}\right) \) gates

\( M \) and \( N \): strongly computes any Boolean function with \( n \) Boolean arguments \( x_0, x_1, \cdots, x_{r-1}, z_0, z_1, \cdots, z_{a-1} \).
Bounded redundancy for Boolean functions

Implication of Theorem 4.3

- [Muller, “Complexity in Electronic Switching Circuits”, 1956]: “Almost all” Boolean functions of $n$ Boolean arguments are computed only by noiseless networks with $\Omega \left(2^n/n\right)$ gates
- “Almost all” Boolean functions have bounded redundancy.

Set of Boolean linear functions

- A set of $m$ Boolean functions $f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)$ is linear if each of the functions is the sum (modulo 2) of some subset of the $n$ Boolean arguments $x_1, \ldots, x_n$.
- “Almost all” sets of $n$ linear functions of $n$ Boolean arguments have bounded redundancy.
  - Similar approach
  - Theorem 4.4
Further readings...

- N. Pippenger, “Reliable computation by formulas in the presence of noise”, 1988
- T. Feder, “Reliable computation by networks in the presence of noise”, 1989