Probabilistic Distributed Computability

E. Godard

"Classical" Impossibility Results in Anonymous Networks
Sources for this talk

Classical sources
- G. Tel 1994 *Intro. to Distributed Algorithms*
- D. Angluin STOC’80 *Local and global properties in networks of processors*
- Probabilistic algorithms from the 80s

More recently: deterministic computability
- classical tools for impossibility proofs
- Mazurkiewicz’ algorithm
Deterministic distributed computability in anonymous networks is related to
Deterministic distributed computability in anonymous networks is related to

- symmetry breaking
- coverings
Deterministic distributed computability in anonymous networks is related to

1. symmetry breaking  
   *coverings*

2. termination detection  
   *quasi-coverings*
Deterministic distributed computability in anonymous networks is related to

1. symmetry breaking
   *coverings*

2. termination detection
   *quasi-coverings*
Deterministic distributed computability in anonymous networks is related to

1. symmetry breaking
   coverings
2. termination detection
   quasi-coverings

What about probabilistic algorithms?
It would certainly help to solve symmetry breaking...
Deterministic distributed computability in anonymous networks is related to

1. symmetry breaking
   *coverings*

2. termination detection
   *quasi-coverings*

What about probabilistic algorithms?
It would certainly help to solve symmetry breaking... Not always...
Not in this Talk

Probabilistic reduction of models
Complexity (rounds, messages, bits)
Terminology: Probability comes in different flavors

A distributed algorithm is probabilistic if nodes have access to a bounded number of *random bits* at each step.

**Sherwood algorithm**
- Algorithm always terminates,
- Result is always correct.

**Las Vegas algorithm**
- Algorithm terminates with probability $0 < p \leq 1$,
- Result is *always correct*.

**Monte Carlo algorithm**
- Algorithm always terminates,
- Result is correct with probability $0 < p \leq 1$. 
A distributed algorithm is probabilistic if nodes have access to a bounded number of random bits at each step.

**Sherwood algorithm**
- algorithm always terminates,
- result is always correct.

**Las Vegas algorithm**
- algorithm terminates with probability $0 < p \leq 1$,
- result is always correct.

**Monte Carlo algorithm**
- algorithm always terminates,
- result is correct with probability $0 < p \leq 1$.

**Very High Probability**
- algorithm always terminates,
- result is incorrect with probability $o\left(\frac{1}{n^c}\right)$, $\forall c \geq 1$. 
A distributed task is $(\Delta, \mathcal{F})$ where

- **specification**: $\Delta \subseteq \mathcal{F} \times \mathcal{F}$, a graph relabelling relation
- **domain**: $\mathcal{F}$, a labelled graph family
A distributed task is $(\Delta, \mathcal{F})$ where

- **specification**: $\Delta \subset \mathcal{F} \times \mathcal{F}$, a graph relabelling relation
- **domain**: $\mathcal{F}$, a labelled graph family

The Election Problem:

- output labels are ELECTED and NONELECTED,
- exactly one node is labelled ELECTED.
Model

- **anonymous**: non *unique* names (i.e. pseudonymous)
- **termination detection**:
  - explicit termination / implicit termination
- any topology / minimal topology
- **no dynamic fault**
Anonymous: non unique names (i.e. pseudonymous)

Termination detection:
- explicit termination / implicit termination
- any topology / minimal topology
- no dynamic fault
- mainly synchronous message passing (but results apply to any model)
**Initial Knowledge**

*Initial knowledge* is a uniform information available (in the initial label) about the structure of the graph. We will consider three types (that represents all situations)

- **None zero knowledge**

- **Bounding for quasi-coverings** the initial information *induces* a family which has the bounded radius quasi-covering property

  Example: bounds on the size, diameter, maximal multiplicity of names, ...

- **Topology** maximal information, giving only the size is enough.
Distributed Computability and the Election Problem

For deterministic algorithms, Election can be solved on (symmetric covering) \textbf{minimal graphs}. 
Distributed Computability and the Election Problem

For deterministic algorithms, Election can be solved on (symmetric covering) **minimal graphs**. General computability results were inspired by studies for the Election problem

- Yamashita & Kameda
- Boldi & Vigna
- Métivier, Chalopin & Godard
- ...

Classical Objective for this Talk

When is Election solvable?
Classical Objective for this Talk

When is Election solvable? For all graphs: universal algorithm.
When is Election solvable? For all graphs: universal algorithm.

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Determ.</th>
<th>Las Vegas</th>
<th>v.h.p</th>
<th>Monte Carlo</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
</tr>
<tr>
<td>Bound</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
</tr>
<tr>
<td>Topology</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
</tr>
</tbody>
</table>
## When is Election solvable?

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Determ.</th>
<th>Las Vegas</th>
<th>v.h.p</th>
<th>Monte Carlo</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>✗</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Bound</td>
<td>✗</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Topology</td>
<td>✗</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Theorem [Angluin80, IR90, Tel94]

It is impossible to solve Election with a universal Las Vegas algorithm.
No Knowledge : First Impossibility Result

Theorem [Angluin80, IR90, Tel94]

It is impossible to solve Election with a universal *Las Vegas* algorithm.

**Proof.** Probabilistic version of the Lifting Lemma.

\[
\begin{array}{c}
G \\
\text{covering} \downarrow \\
H
\end{array}
\]
Theorem [Angluin80, IR90, Tel94]

It is impossible to solve Election with a universal Las Vegas algorithm.

Proof. Probabilistic version of the Lifting Lemma.
Theorem [Angluin80, IR90, Tel94]

It is impossible to solve Election with a universal Las Vegas algorithm.

Proof. Probabilistic version of the Lifting Lemma.

\[ G \xrightarrow{\text{covering}} H \xrightarrow{A^* \text{ with proba. } p > 0} H' \]
Theorem [Angluin80, IR90, Tel94]

It is impossible to solve Election with a universal *Las Vegas* algorithm.

**Proof.** Probabilistic version of the Lifting Lemma.

\[
\begin{align*}
G & \xrightarrow{A^*} G' \\
& \text{with proba. } p' > 0 \\
H & \xrightarrow{A^*} H' \\
& \text{with proba. } p > 0
\end{align*}
\]
What about Monte-Carlo?

Theorem [from IR90, Tel94]

It is impossible to solve Election with a universal *Monte Carlo* algorithm with $\varepsilon > 0$. 
What about Monte-Carlo?

Theorem [from IR90, Tel94]
It is impossible to solve Election with a universal Monte Carlo algorithm with $\varepsilon > 0$.

Proof.

- Consider an algorithm $A$ with success probability $\varepsilon > 0$,
- Consider $H$ a network of size $n$
- Consider a terminated execution of $r$ rounds and probability $q$,
- We choose a quasi-covering of $H$ of radius $r + 2n$ and we apply the probabilistic quasi-lifting lemma.
End of the Proof

\[ \text{quasi-covering of radius } R \]

\[ \downarrow \]

\[ \text{H} \]
End of the Proof

\[ \text{quasi-covering of radius } R \]

\[ \begin{array}{c}
G \\
\downarrow \\
H \\
\end{array} \xrightarrow{\mathcal{A}^r} \begin{array}{c}
H' \\
\text{with proba. } q > 0
\end{array} \]

The remaining quasi-covering has at least 2 sheets, so we have 2 ELECTED nodes. It is a failed run.

What is the probability \( q' \)?

Very small but positive.

Consider an integer \( R \) such that \((1 - q') R < \epsilon\) and a quasi-covering of radius \( R (r + 2n) \).

We have \( R \) disjoint \( r + 2n \) quasi-coverings, then the probability of having at least a failed run is \( > 1 - \epsilon \).

Contradiction.
End of the Proof

\[ G \xrightarrow{A^*} G' \quad \text{with proba. } q' > 0 \]

\[ H \xrightarrow{A^r} H' \quad \text{with proba. } q > 0 \]

quasi-covering of radius \( R \)

quasi-covering of radius \( 2n \)
End of the Proof

The "remaining" quasi-covering has at least 2 sheets, so we have 2 ELECTED nodes. It is a failed run.

What is the probability $q'$? very small but positive.
End of the Proof

- The "remaining" quasi-covering has at least 2 sheets, so we have 2 ELECTED nodes. It is a **failed** run.
- What is the probability $q'$? *very small* but positive.
- Consider an integer $R$ such that $(1 - q')^R < \varepsilon$ and a quasi-covering of radius $R(r + 2n)$.
- We have $R$ disjoint $r + 2n$ quasi-coverings,
- Then the probability of having at least a failed run is $> 1 - \varepsilon$. **Contradiction.**
Corollary

It is impossible to solve Election with \( v.h.p. \) with a universal algorithm.
### When is Election Solvable: No Information

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Determ.</th>
<th>Las Vegas</th>
<th>v.h.p</th>
<th>Monte Carlo</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Bound</td>
<td></td>
<td>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Topology</td>
<td></td>
<td>?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

E. Godard (LIF – Université Aix-Marseille)
Theorem [Angluin80][Tel94]

It is impossible to solve Election with a *Las Vegas* algorithm, even knowing a bound.

**Proof.** The lifting lemma also works with bounds knowledge.
## When is Election Solvable: Partial Summary

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Determ.</th>
<th>Las Vegas</th>
<th>v.h.p</th>
<th>Monte Carlo</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td><img src="#" alt="X" /></td>
<td><img src="#" alt="X" /></td>
<td><img src="#" alt="X" /></td>
</tr>
<tr>
<td>Bound</td>
<td><img src="#" alt="X" /></td>
<td><img src="#" alt="X" /></td>
<td><img src="#" alt="?" /></td>
<td><img src="#" alt="?" /></td>
</tr>
<tr>
<td>Topology</td>
<td><img src="#" alt="X" /></td>
<td><img src="#" alt="?" /></td>
<td><img src="#" alt="?" /></td>
<td><img src="#" alt="?" /></td>
</tr>
</tbody>
</table>
Probabilistic Election Algorithms

(1) Itai & Rodeh, : Las Vegas Election Algorithm knowing the size.
(2) Schieber & Schnir : Monte Carlo Algorithm knowing a bound
(3) [CGPS97] : Election with v.h.p (with 1 random bit) knowing the size, can be extended to knowing a bound with more bits
### Election: Knowledge vs Prob. Strength

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Determ.</th>
<th>Las Vegas</th>
<th>v.h.p</th>
<th>Monte Carlo</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Bound</td>
<td>❌</td>
<td>❌</td>
<td>✔️</td>
<td>✔️ (2)</td>
</tr>
<tr>
<td>Topology</td>
<td>❌</td>
<td>✔️ (1)</td>
<td>✔️</td>
<td>✔️ (3)</td>
</tr>
</tbody>
</table>
### Computability: Knowledge vs Prob. Strength

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Determ.</th>
<th>Las Vegas</th>
<th>v.h.p</th>
<th>Monte Carlo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bound</td>
<td>![Star]</td>
<td>![Star]</td>
<td>![Star]</td>
<td>![Star]</td>
</tr>
<tr>
<td>Topology</td>
<td>![Star]</td>
<td>![Star]</td>
<td>![Star]</td>
<td>![Star]</td>
</tr>
</tbody>
</table>

- **Covering-stable** in $\mathcal{F}_\downarrow$
- **Quasi-covering-stable**
- **Everything**

---

E. Godard (LIF – Université Aix-Marseille)

Probabilistic Distributed Computability

Displexity - 05/04 La Rochelle
Reminder

Given a computation model on labelled graphs $G$, $\ast$-coverings defines

**Extended graph family** $G \downarrow$ graphs with possibly self-loops and multi-arcs

**Homomorphisms** the $\ast$-coverings
Reminder

Given a computation model on labelled graphs $G$, *-coverings defines

Extended graph family $G↓$ graphs with possibly self-loops and multi-arcs

Homomorphisms the *-coverings

$$\begin{align*}
G \in G \\
\quad \downarrow \\
*\text{-covering} \\
H \in G \\
\quad \downarrow \\
*\text{-covering} \\
K \in G↓
\end{align*}$$
Idea of the Proofs

Given a labelled graphs family $\mathcal{F}$, we have to consider

Extended graph family $\mathcal{F} \downarrow$ graphs covered by graphs of $\mathcal{F}$ (with possibly self-loops and multi-arcs)

Homomorphisms the $\ast$-coverings

\[
G \in \mathcal{F} \\
\downarrow \text{*-covering} \quad \downarrow \\
H \in \mathcal{F} \\
\downarrow \text{*-covering} \\
K \in \mathcal{F} \downarrow
\]

With probabilistic correctness, $\mathcal{F} \downarrow = \mathcal{F}$. 

E. Godard (LIF – Université Aix-Marseille)
Idea of the Proofs

Given a labelled graphs family $\mathcal{F}$, we have to consider

Extended graph family $\mathcal{F} \downarrow$ graphs covered by graphs of $\mathcal{F}$ (with possibly self-loops and multi-arcs)

Homomorphisms the $\ast$-coverings

\[
\begin{align*}
G & \in \mathcal{F} \\
\ast\text{-covering} & \downarrow \\
H & \in \mathcal{F} \\
\ast\text{-covering} & \downarrow \\
K & \in \mathcal{F} \downarrow \\
\end{align*}
\]

With probabilistic correctness, $\mathcal{F} \downarrow = \mathcal{F}$. 
Quasi-covering Stable Tasks

Definition r-lifting closed

Given a function $r : (\mathcal{G}, .) \rightarrow \mathbb{N}$ and a function $f : (\mathcal{G}, .) \rightarrow \Lambda$, the function $f$ is $r$—lifting closed if for all $G, H$ such that $G$ is a quasi-covering of $H$, of center $v_0$ and of radius $R$ via $\gamma$.

If $R \geq \min\{r(G, v_0), r(H, \gamma(v_0))\}$, then $f(G, v_0) = f(H, \gamma(v_0))$.

A task $\Delta$ is quasi-covering stable if there exists $r$ and $f$ such that

- $f$ and $r$ are $r$—lifting closed
- $f$ computes $\Delta$
### Exercice : Election on Minimal Graphs

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Determ.</th>
<th>Las Vegas</th>
<th>v.h.p</th>
<th>Monte Carlo</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
</tr>
<tr>
<td>Bound</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
</tr>
<tr>
<td>Topology</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
<td>![Question Mark]</td>
</tr>
</tbody>
</table>
### Hint: Election on Minimal Graphs

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Determ.</th>
<th>Las Vegas</th>
<th>v.h.p</th>
<th>Monte Carlo</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>✗</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Bound</td>
<td>✓ (4)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Topology</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

(4) G. & Métivier, Fossacs’02  
Chalopin, G. & Métivier, Dist. Comp.’12
The quasi-lifting lemma applies also (especially) to minimal graphs.
Improving Computability by less requirements:

**Implicit Termination**

It is possible *temporarily* to have more than one ELECTED node.
### Election: Implicit Termination on All Graphs

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Determ.</th>
<th>Las Vegas</th>
<th>v.h.p</th>
<th>Monte Carlo</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>❌</td>
<td>❌</td>
<td>❓</td>
<td>✔</td>
</tr>
<tr>
<td>Bound</td>
<td>❌</td>
<td>❌</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Topology</td>
<td>❌</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

(5) Matias & Afek 89

**Remark**: Only the Lifting Lemma can be applied.
## Computability: Implicit Termination

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Determ.</th>
<th>Las Vegas</th>
<th>v.h.p</th>
<th>Monte Carlo</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>![Star]</td>
<td>![Star]</td>
<td>![Question Mark]</td>
<td>![Green Circle]</td>
</tr>
<tr>
<td>Bound</td>
<td>![Star]</td>
<td>![Star]</td>
<td>![Green Circle]</td>
<td>![Green Circle]</td>
</tr>
<tr>
<td>Topology</td>
<td>![Star]</td>
<td>![Green Circle]</td>
<td>![Green Circle]</td>
<td>![Green Circle]</td>
</tr>
</tbody>
</table>

E. Godard (LIF – Université Aix-Marseille)
Exercice : Implicit Termination and Minimal Graphs
**Exercise: Implicit Termination and Minimal Graphs**

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Determ.</th>
<th>Las Vegas</th>
<th>v.h.p</th>
<th>Monte Carlo</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>☑️</td>
<td>☑️</td>
<td>☑️</td>
<td>☑️</td>
</tr>
<tr>
<td>Bound</td>
<td>☑️</td>
<td>☑️</td>
<td>☑️</td>
<td>☑️</td>
</tr>
<tr>
<td>Topology</td>
<td>☑️</td>
<td>☑️</td>
<td>☑️</td>
<td>☑️</td>
</tr>
</tbody>
</table>

**Remark:** Using Mazurkiewicz’ algorithm.
Definition [Gouda WSS’01]
An execution is **Gouda fair** if a global transition that is *possible* infinitely often occurs infinitely often.

**Remark.** Fairness for population protocol.
Theorem

Given a stabilizing tasks $\Delta$, there is a Las Vegas algorithm for $\Delta$ only if there is a deterministic algorithm for $\Delta$ correct for a Gouda fair scheduler.

Idea of the proof. A Gouda fair scheduler avoids the lifting lemma, that can occurs for schedules that have measure 0.
Given maximal information, everything can be computed with a probabilistic algorithm,

With no information, all that can be obtained is v.h.p. **without** explicit termination,

With **implicit termination**, every task $\Delta$ is universally solvable on the family where $\Delta$ is solvable knowing the topology.
Questions?

Thank you very much.
References

[Angluin80] Angluin, D. *Local and Global Properties in Networks of Processors* STOC’80


[CGM08] Chalopin, J. and Godard, E. and Métivier, Y. *Local Terminations and Distributed Computability in Anonymous Networks*, DISC 2008

References


