Distributed decision

Laurent Feuilloley
Université Paris Diderot

based on joint works with Pierre Fraigniaud,
Juho Hirvonen, Ami Paz, and Mor Perry

2017
Distributed decision
Building vs. deciding

Yes/No
Distributed languages

Context:

- Communication graph $G$
- Node inputs, $x : v \mapsto x(v)$

A language is a set $\{(G, x)\}$. 
Properly-colored graphs

\[ L = \{(G, x) \text{ s.t. } x \text{ is a proper coloring of } G\} \]
Spanning forest

\[ \mathcal{L} = \{ (G, x) \text{ s.t. } x \text{ describes a spanning forest of } G \} \]
Spanning forest

\[ \mathcal{L} = \{ (G, x) \text{ s.t. } x \text{ describes a spanning forest of } G \} \]
Decision mechanism

Every node:
- gathers its 1-neighbourhood
- outputs a local decision accept or reject.
Decision mechanism

Every node:
- gathers its 1-neighbourhood
- outputs a local decision: accept or reject.
Decision mechanism

Every node:

- gathers its 1-neighbourhood
- outputs a local decision accept or reject.
Decision mechanism

\((G, x)\) is accepted if all node accept.
Decision mechanism

$(G, x)$ is rejected if at least one node rejects.
Properly-colored graphs
Properly-colored graphs
Properly-colored graphs
Properly-colored graphs
Properly-colored graphs
Properly-colored graphs
Properly-colored graphs
Properly-colored graphs
Spanning forest
Spanning forest
Spanning forest
Spanning forest
Spanning forest
Spanning forest
Proof-labeling schemes

Distributed non-determinism
Proof-labeling schemes
Proof-labeling schemes
Proof-labeling schemes
Proof-labeling schemes

Given a proof-labeling scheme for $\mathcal{L}$:

For all $(G, x)$:

- If $(G, x) \in \mathcal{L}$:
  \[ \exists c \text{ s.t. } (G, x, c) \text{ is accepted.} \]

- If $(G, x) \notin \mathcal{L}$:
  \[ \forall c, (G, x, c) \text{ is rejected.} \]
PLS on spanning forest
PLS on spanning forest
PLS on spanning forest
PLS on spanning forest
PLS on spanning forest
Motivations
Self-stabilizing algorithms
Self-stabilizing algorithms
Self-stabilizing algorithms
Self-stabilizing algorithms
Self-stabilizing algorithms
Self-stabilizing algorithms
Self-stabilizing algorithms
Self-stabilizing algorithms
Untrusted oracle

External computation
Untrusted oracle

External computation
Measuring locality via proof sizes

- **$O(1)$**
  - 3-colored
  - Spanning tree

- **$\Theta(\log n)$**
  - Minimum spanning tree

- **$\Theta(n^2)$**
  - Non-3-colorable
  - Symmetry
Unifying models
Unifying models
New works

Proof-labeling schemes

- Alternation
- Change prover
- Global proofs
- Change prover
- Change verifier
- Change Decision
- Stronger rejection
- Larger radius
Stronger rejection

a.k.a.

Error-sensitivity of proof-labeling schemes
One node to reject
More nodes to reject
Error-sensitivity

A PLS is error-sensitive if the number of rejecting nodes grows linearly with the distance.
Characterization

A language $\mathcal{L}$ admits an error-sensitive PLS $\iff \mathcal{L}$ is locally stable.
Local stability

Hybridization

\[ \in \mathcal{L} \]
Landscape

- $O(1)$
- $\Theta(\log n)$
- Minimum spanning tree
- Spanning tree
- 3-colorable
- 3-colored

No error-sensitive PLS
- ST with pointers
- MST with pointers
- Symmetry
Larger radius
Decision mechanism

Every node:

- gathers its 1-neighbourhood
- outputs a local decision accept or reject.
Larger radius
Larger radius
Smaller proofs?

What trade-offs between the radius \( t \) and the certificate size?

- Can we always get \( s_t(n) = s_1(n)/t \) ?
- When can we get \( s_t(n) = s_1(n)/b(t) \) ?
Spreading uniform proofs
Spreading uniform proofs
Spreading uniform proofs
Spreading uniform proofs
Global proofs
A global proof
Basic inequalities

For a fixed language.

- Local(n) : the optimal size for local proofs.
- Global(n) : the optimal size for global proofs.

\[
\text{Local}(n) \leq \text{Global}(n) \leq n \times [\text{Local}(n) + \log n]
\]
Two selection problems

- AMOS: at most one node is selected
- ALOS: at least one node is selected

\[ \text{AMOS} \cap \text{ALOS} = \text{‘Leader elected’} \]
Two selection problems

<table>
<thead>
<tr>
<th></th>
<th>Local(n)</th>
<th>Global(n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMOS</td>
<td>$\log n$</td>
<td>$\log n$</td>
</tr>
<tr>
<td>ALOS</td>
<td>$\log n$</td>
<td>$n \log n$</td>
</tr>
</tbody>
</table>
Alternation
Reversing decision

\((G, x)\) is rejected if at least one node rejects.
Reversing decision

- $\Theta(\log n)$
- Minimum spanning tree
- Spanning tree
- 3-colorable
- 3-colored
- Non-3-colorable
- Symmetry

$O(1)$ $\Theta(n^2)$
Prover vs. disprover
Remember PLS

Given a proof-labeling scheme for $\mathcal{L}$:

For all $(G, x)$:

- If $(G, x) \in \mathcal{L}$:
  $\exists c$ s.t. $(G, x, c)$ is accepted.

- If $(G, x) \notin \mathcal{L}$:
  $\forall c$, $(G, x, c)$ is rejected.
Disprover-prover scheme

Given a disprover-prover scheme for $\mathcal{L}$:

For all $(G, x)$:

- If $(G, x) \in \mathcal{L}$:
  \[ \forall c_d \exists c_p \text{ s.t. } (G, x, c_d, c_p) \text{ is accepted.} \]

- If $(G, x) \notin \mathcal{L}$:
  \[ \exists c_d \forall c_p, (G, x, c_d, c_p) \text{ is rejected.} \]
Some conversations

- Non-3-colorable: disprover-prover
- Optimal combinatorial solution: disprover-prover
- Symmetry: prover-disprover-prover
- Some language: talk forever.
Conclusion

Playing with non-determinism is:

- useful, to model to different systems,
- a good way to study locality.