Randomized Local Network Computing

Laurent Feuilloley\textsuperscript{1} · Pierre Fraigniaud\textsuperscript{2}

\textsuperscript{1}ENS Cachan · Université Paris Diderot · Aalto University
\textsuperscript{2}Université Paris Diderot

SPAA 2015 · Portland · June 15
A derandomization theorem

Informally our theorem is:

- In a distributed computing model
- If a language can be checked locally with randomization
- Then:
  - If it can be constructed locally with randomization
  - Then it can be constructed locally without randomization
LOCAL model

- A network of machines
- Every vertex has a unique identifier

![Diagram of a network with nodes labeled ID: 235, Computation, and Communication]
LOCAL model

- A network of machines
- Every vertex has a unique identifier

ID: 235

Computational power = ∞
No failure

Communication Bandwidth = ∞
LOCAL model

First point of view: minimize the number of rounds
Second point of view: local computation

max degree $= \Delta$
Theorem

- In the LOCAL model
- If a language can be checked locally with randomization
- Then:
  - If it can be constructed locally with randomization
  - Then it can be constructed locally without randomization
Construction vs Decision

Example: $(\Delta + 1)$-coloring

**Construction** from a **global** perspective.
Construction vs Decision

Construction from a local perspective.
Construction vs Decision

Theorem (Linial’92):

Constructing a \((\Delta + 1)\)-colouring requires \(\Omega(\log^* n)\) rounds.
Construction vs Decision

Decision from a local perspective
Construction vs Decision

Decision from a local perspective
Construction vs Decision

Decision from a global perspective
Construction vs Decision

Decision from a global perspective
Theorem

• In the LOCAL model
• If a language can be checked \textit{locally} with randomization
• Then:
  • If it can be constructed \textit{locally} with randomization
  • Then it can be constructed \textit{locally} without randomization
Locally?

Here locally means **constant** number of rounds

Coloring verification can be done locally $\rightarrow 1$ round, but coloring construction cannot $\rightarrow \log^* n$ rounds.
Locally?

What can be constructed locally?

→ Weak coloring, fractional coloring, and some approximations
Theorem

- In the LOCAL model
- If a **language** can be checked in $O(1)$ rounds, with **randomization**
- Then:
  - If it can be constructed in $O(1)$ rounds with **randomization**
  - Then it can be constructed in $O(1)$ rounds without randomization
Languages and classes

- a **language**: is a set \( \{(G, x) \text{ satisfying a property } P \} \)
- A **class** is a set of languages

\[ \rightarrow LD = \text{the languages that can be checked in constant time deterministically.} \]
Languages and classes

\textbf{BPLD} = the languages that can be checked in constant time using \textit{randomization}.

More precisely:
there exists a checker, and \( p \in \left( \frac{1}{2}, 1 \right] \) s.t.:

- If \((G, x) \in \mathcal{L}\), then \(\Pr[\text{all nodes accept}] \geq p\)
- If \((G, x) \notin \mathcal{L}\) then \(\Pr[\text{a node rejects}] \geq p\)
$f$-resilient tasks and BPLD

\[ \mathcal{L}_f = \mathcal{L} \in \text{LD} + \text{errors accepted on } f \text{ nodes.} \]
$f$-resilient tasks and BPLD

Coloring with one bad edge is ok, but not more.
$f$-resilient tasks and BPLD

Strategy:
- If the coloring is good, accept
- If the coloring is bad, reject with probability $q$
Back to the derandomization theorem

More formally the theorem is:

- In the LOCAL model
- If $\mathcal{L} \in \text{BPLD}$
- Then:
  - If $\mathcal{L}$ can be constructed in $O(1)$ rounds with randomization
  - Then it can be constructed in $O(1)$ rounds deterministically
A glimpse of the proof

Two main steps:

- Using Ramsey theory to reduce to a special case (from Naor-Stockmeyer ’93)
- Proving that locality prevent weird correlations
Further works

- LD $\rightarrow$ BPLD $\rightarrow$ ?
- Get a better understanding of randomization in network distributed computing.
Further works

- LD $\rightarrow$ BPLD $\rightarrow$ ?
- Get a better understanding of randomization in network distributed computing.

Thank you!