NETWORKS OF PICTURE PROCESSORS

Victor Mitrana

joint work with

Henning Bordihn (Potsdam)
Paolo Bottoni (Rome)
Anna Labella (Rome)
CONTENTS

General idea
Sources of inspiration (DNA as a computing tool)
Informal description of the model
How the model computes
Controlling the communication
Types of processors
Advantages and drawbacks
Discussions (questions, comments, suggestions, solutions, etc.)
Is the classic architecture sufficient?

Make the computation in parallel

Distribute the computation

Approximate the solution

Change the “ingredients”: natural computing
- Genetic algorithms, neural networks
- Quantum computing
- Molecular computing
and many others
**Distributed computing**: networks of computer-like devices that can exchange large messages with their neighbors and perform arbitrary local computations.
- Processors having limited memory and computational abilities.
- Processors having no a priori knowledge of their location.
- Each processor has only a limited, incomplete view of the system.

**AMORPHOUS COMPUTING**
A FUNNY (IM)POSSIBLE WAY OF SOLVING PROBLEMS
A set of nodes that are connected: a network
Each node: simple processor
Three distinguished nodes: In, Halt and Accept

The edges: bidirectional communication channels
Information: strings, pictures, graphs
Restricted communication by: input/output filters.

Input: a picture in In.
Computation: Processing, Communication, .....  
Halting: a picture enters Halt
Acceptance: when the computation halts, Accept is nonempty
**Goal:** To apply distributed computing methods to networks of sub-microprocessor devices, e.g., biological cellular networks or networks of nano-devices.

**Question:** do tiny bio/nano nodes “compute” and/or “communicate” essentially the same as a computer?

**Our attempt:** Although the computation and communication capabilities of each individual device in the new model are, by design, much weaker than those of a computer, we show that some of the most important and extensively studied distributed computing problems can still be solved efficiently.

Theorists: **YES, WE CAN!**
Engineers: ?
Biologists: ?
IMAGES AS RECTANGULAR PICTURES

Stored image consists of two-dimensional array of pixels (picture elements):

Many low-level image-processing operations assume monochrome images and refer to pixels as having gray level values or intensities.
\[ \pi = \begin{array}{cccc}
  a & b & b & a \\
  b & b & a & a \\
  c & b & a & a \\
  b & b & a & a \\
\end{array} \]

\[ \rho = \begin{array}{ccccccc}
  a & b & b & b & b & a & b \\
  b & b & a & b & a & a & b \\
  c & b & a & b & a & a & c \\
  b & b & a & b & a & a & b \\
\end{array} \]

\[ \pi \odot \rho = \begin{array}{ccccccc}
  a & b & b & a & c & a & b \\
  b & b & a & a & b & b & a \\
  c & b & a & a & a & c & b \\
  b & b & a & a & a & b & b \\
\end{array} \]

\[ \pi \supseteq \rho = \begin{array}{ccccccc}
  a & b & b & b & b & a & b \\
  b & b & a & b & a & a & b \\
  c & b & a & b & a & a & c \\
  b & b & a & b & a & a & b \\
\end{array} \]
**EVOLUTIONARY OPERATIONS AND ACTIONS:**

CSubstitution: $a \rightarrow b (|)$

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>b</td>
<td>a</td>
<td>c</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td></td>
</tr>
</tbody>
</table>

$\sigma^*(\pi)$

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
<td>a</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>a</td>
<td>b</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>b</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>
**OPERATIONS ON PICTURES**

**EVOLUTIONARY OPERATIONS AND ACTIONS:**

CSubstitution: $a \rightarrow b \ (||)$  

\[
\begin{array}{cccc}
  a & b & b & a \\
  b & b & a & a \\
  c & b & a & a \\
  b & b & a & a \\
\end{array}
\]

\[
\begin{array}{cccc}
  b & b & b & a \\
  b & b & a & a \\
  c & b & a & a \\
  b & b & a & a \\
\end{array}
\]
**EVOLUTIONARY OPERATIONS AND ACTIONS:**

**CSubstitution:** $a \rightarrow b$  

\[
\begin{array}{cccc}
  a & b & b & a & c \\
  b & b & a & a & b \\
  c & b & a & a & a \\
  b & b & a & a & a \\
\end{array}
\]

\[
\sigma \rightarrow (\pi)
\]

\[
\begin{array}{cccc}
  a & b & b & a & c \\
  b & b & a & a & b \\
  c & b & a & a & b \\
  b & b & a & a & a \\
\end{array}
\]

\[
\begin{array}{cccc}
  a & b & b & a & c \\
  b & b & a & a & b \\
  c & b & a & a & a \\
  b & b & a & a & a \\
\end{array}
\]

\[
\begin{array}{cccc}
  a & b & b & a & c \\
  b & b & a & a & b \\
  c & b & a & a & a \\
  b & b & a & a & b \\
\end{array}
\]
EVOLUTIONARY OPERATIONS AND ACTIONS:

RDeletion: $a \rightarrow \lambda (-)$

$$\sigma^*(\pi)$$

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>b</td>
<td>a</td>
<td>c</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td></td>
</tr>
</tbody>
</table>
**OPERATIONS ON PICTURES**

**EVOLUTIONARY OPERATIONS AND ACTIONS:**

RDletion: $a \rightarrow \lambda (-)$

\[
\begin{array}{cccc}
  a & b & b & a & c \\
  b & b & a & a & b \\
  c & b & a & a & a \\
  b & b & a & a & a \\
\end{array}
\]

$\sigma^\uparrow(\pi)$

\[
\begin{array}{cccc}
  b & b & a & a & b \\
  c & b & a & a & a \\
  b & b & a & a & a \\
\end{array}
\]
**OPERATIONS ON PICTURES**

**EVOLUTIONARY OPERATIONS AND ACTIONS:**

**RDeletion:** $a \rightarrow \lambda (-) \quad \sigma \downarrow (\pi)$

$$
\begin{array}{cccc}
  a & b & b & a & c \\
  b & b & a & a & b \\
  c & b & a & a & a \\
  b & b & a & a & a \\
\end{array}
$$

$$
\begin{array}{cccc}
  a & b & b & a & c \\
  b & b & a & a & b \\
  c & b & a & a & a \\
\end{array}
$$
FORMAL DEFINITIONS

FILTERS:

\( \varphi^{(s)}(w; P, F) \equiv P \subseteq \text{alph}(w) \quad \land \quad F \cap \text{alph}(w) = \emptyset. \)

\( \varphi^{(w)}(w; P, F) \equiv \text{alph}(w) \cap P \neq \emptyset \quad \land \quad F \cap \text{alph}(w) = \emptyset. \)

\( \varphi^\beta(L, P, F) = \{ w \in L : \varphi^\beta(w; P, F) \}. \)
FORMAL DEFINITIONS

EVOLUTIONARY PICTURE PROCESSOR: \((M, PI, FI, PO, FO)\)

ANEPP: \(\Gamma=(V, U, G, N, \alpha, \beta, In, Halt, Accept)\)

\[ G=(X_G, E_G) : \text{underlying graph structure} \]
\[ N: X_G \rightarrow EP_v : \text{associated picture processors} \]
\[ \alpha : X_G \rightarrow \{*, \leftarrow, \rightarrow, \downarrow, \uparrow\} : \text{action mode} \]
\[ \beta : X_G \rightarrow \{s, w\} : \text{filter type} \]

\[ \rho_x(.)=\varphi^{\beta(x)}(.; PI_x, FI_x) : \text{input filter} \]
\[ \tau_x(.)=\varphi^{\beta(x)}(.; PO_x, FO_x) : \text{output filter} \]
WORKING MODE

Evolutionary step:

\[ C \Rightarrow C', \text{ iff } C'(x) = M_x \alpha(x)(C(x)) \]

Communication step:

\[ C \gg C' \text{ iff } C'(x) = (C(x) - \tau_x(C(x))) \cup \bigcup_{\{x,y\} \in EG} (\tau_y(C(y)) \cap \rho_x(C(y))) \]
LOCAL PICTURE LANGUAGE

LOC = the class of local picture languages

\[ \Theta = \left\{ \begin{array}{cccccccc}
1 & 0 & 0 & 0 & \# & 0 & \# & 1 & \#
\end{array} \right\} \]
Th. [BLM – Fundamenta Informaticae 2014]
1. The complement of every local language can be weakly accepted by an ANEPP.
2. There exist non-recognizable languages which can be accepted by ANEPPs.
Problem. Given a pattern $\pi$ and a picture $\theta$ is $\pi$ a subpicture of $\theta$?
Th. [BBLM – TPNC 2014]

1. Let $\pi$ be a picture of size $(k; n)$ for some $1 \leq k \leq 3$ and $n \geq 1$. The language $\{\pi\}$ can be accepted by an ANEPP.

2. Given a finite set $F$ of patterns of size $(k; l)$ and $(l; k)$ for all $1 \leq k \leq 3$ and $l \geq 1$, the pattern matching problem with patterns from $F$ can be solved by ANEPPs in $O(n+m+l)$ computational (processing and communication) steps.
FURTHER OPERATIONS ON PICTURES

MASK and UNMASK:

mask↓

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>a</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>
FURTHER OPERATIONS ON PICTURES

MASK and UNMASK:

mask↓

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>
**FURTHER OPERATIONS ON PICTURES**

```
mask↑
```

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>

```
<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>
```
unmask →

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>
Th. [BBLM – TPNC 2014]

Let $\pi$ be a picture of size $(k; l)$ for some $k, l \geq 1$. The language $\{\pi\}$ can be accepted by an ANPP.
SOLVING 2D PATTERN MATCHING
SOLVING 2D PATTERN MATCHING
THEOREM [BBLM, TPNC 2014]
Given a finite set $F$ of patterns of size $(k,l)$ for any $k,l \geq 1$, the pattern matching problem with patterns from $F$ can be solved by ANPPs in $O(n+m+kl+k)$ computational (processing and communication) steps.
1. Let \((k; l)\) be two positive integers, \(1 \leq k \leq 3\) and \(l \geq 1\). Every \((k; l)\)-local language or \((l; k)\)-local language can be decided by ANEPPs in \(O(n + m + l)\) computational (processing and communication) steps.

2. Let \((k; l)\) be two positive integers. Every \((k; l)\)-local language can be decided by ANEPPs in \(O(n + m + kl)\) computational (processing and communication) steps.
Let \((k; l)\) be two positive integers and \(F\) be a finite set of pictures of size \((k; l)\). The picture language \(F^*\) is the minimal set of pictures such that:

\[ (i) \quad F \subseteq F^*, \]

\[ (ii) \quad \text{If } \pi, \rho \in F^*, \text{ then } \pi \otimes \rho \in F^* \text{ (provided that } \pi \otimes \rho \text{ exists)} \text{ and } \pi \odot \rho \in F^* \text{ (provided that } \pi \odot \rho \text{ exists).} \]

**Th. [BBLM – Soft Computing 2016]**

Let \((k; l)\) be two positive integers and \(F\) be a finite set of pictures of size \((k; l)\). The language \(F^*\) can be decided by ANEPPs in \(O(n + m + kl)\) computational (processing and communication) steps.
Thank You