Networks of Bio-Inspired Processors. An introduction

José M. Sempere
Research Group on Computation Models and Formal Languages
Departmento de Sistemas Informáticos y Computación
Universitat Politècnica de València

Networks of Bio-Inspired Processors. An introduction

A NBP is a computational model which

… is inspired by biological aspects (darwinian evolution, DNA recombination, etc.)

… is computationally complete (it has the computation power of a Turing machine)

… is parallel and distributed

… is universal (allows the interpretation of NBPs as source data)

… solves NP-complete problems in "polynomial" time

Networks of Bio-Inspired Processors. An introduction

	P systems	NBPs
Computationally complete and universal	OK	OK
Parallel and distributed	OK	OK
Works with strings	OK	OK
Hardware implementations	OK+KO	OK + KO
Works with multisets of data	OK	OK
Software simulators	OK	OK + KO
In vitro/in vivo implementations	KO	KO
Efficient solutions to hard problems	OK	OK

From NEPs to P Systems → Evolutionary P systems (Mitrana, Sempere 2009)

From P Systems to NBPs → Open problem

Networks of Bio-inspired Processors

Some bioinspired operators over strings and languages

Insertion Insert a symbol into a string — aaaaa ➔ aabaaa

Deletion Delete a symbol from a string — aabaaa ➔ aaaaa

Substitution (mutation) Substitute a symbol into a string — aaaaa ➔ aabaa

Splicing Splicing rules $r=(u_1\#u_2\$v_1\#v_2)$ — r=(a#a$b#b) (abcdaa,bbabcd) ➔ (abcdababcd,ba)

Crossover Full massive splicing with empty context — aa ▹◃bb ➔ λ, bb, abb, aabb, ab, aab, …

Hairpin completion Hairpin completion from folded strings

Superposition Complementarity completion from double stranded strings

loop and double loop recombination DNA recombination based on gene assembly

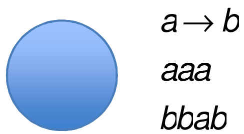
inversion, duplication and transposition DNA fragments modification (operations on substrings)

… etc, etc.

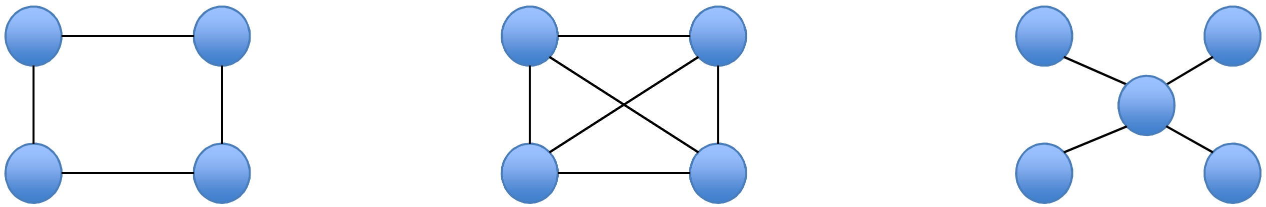
Networks of Bio-inspired Processors

The ingredients to define a Network of Bioinspired Processors

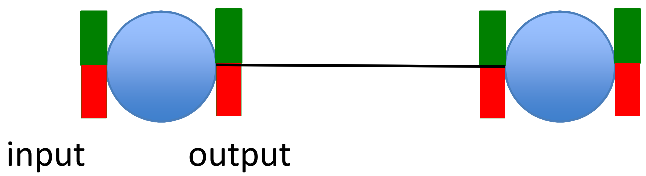
A finite set of processors that apply operations over strings which have been inspired by biomolecular functions and operations in the nature. The processors work with a multiset of strings.

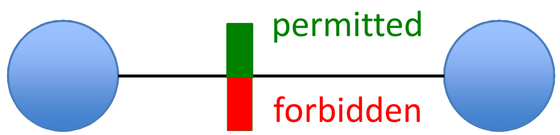


A connection topology between processors in the form of a network.



A set of (input/output) filters which can be attached to the processors or to the connections.





Networks of Bio-Inspired Processors

Accepting Networks of Evolutionary Processors

An evolutionary processor over *V* is a 5-tuple *(M,PI,FI,PO,FO)*,where:

Either $M \subseteq Sub_V$ or $M \subseteq Del_V$ or $M \subseteq Ins_V$
The set M represents the set of evolutionary rules of the processor.

$PI,FI \subseteq V$ are the input permitting/forbidding contexts of the processor

$PO,FO \subseteq V$ are the output permitting/forbidding contexts of the processor
(with $PI \cap FI = \emptyset$ and $PO \cap FO = \emptyset$)

We can define the following predicated for the filters

$$rc_s(z, P, F) \equiv [P \subseteq alph(z)] \wedge [F \cap alph(z) = \emptyset]$$

$$rc_w(z, P, F) \equiv [alph(z) \cap P = \emptyset] \wedge [F \cap alph(z) = \emptyset]$$

Networks of Bio-Inspired Processors

Accepting Networks of Evolutionary Processors

$$\Gamma = (V, U, G, N, \alpha, \beta, x_I, x_O)$$

where

V and U are the input and network alphabets

$G=(X_G,E_G)$ is an undirected graph without loops

N: $X_G \rightarrow EP_U$ associates an evolutionary processor to every node in G

α: $X_G \rightarrow \{l,r,*\}$ associates an action mode to every node (Hybrid networks)

β: $X_G \rightarrow \{s,w\}$ associates a filter predicate to every node

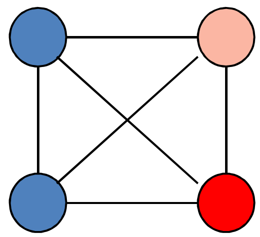
x_I, x_O are the input and output nodes

Networks of Bio-Inspired Processors

Accepting Networks of Evolutionary Processors

$$\Gamma = (V, U, G, N, \alpha, \beta, x_I, x_O)$$

How does the network work ?



(I) Evolutionary steps

$$C_i \Rightarrow C_{i+1}$$

- *Every rule that can be applied is massively applied*
- *No competition between rules. All the rules are applied by using different copies*

(II) Communication steps

$$C_i \mapsto C_{i+1}$$

- *Every processor sends all the filtered strings to its neighbours*
- *Every processor receives and stores filtered strings*
- *Strings that are sent but not received are lost*

(III) Network at work

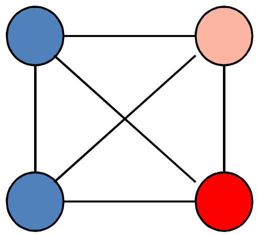
$$C_0 \Rightarrow C_1 \mapsto C_2 \Rightarrow C_3 \mapsto C_4 \ldots$$

Networks of Bio-Inspired Processors

Accepting Networks of Evolutionary Processors

$$\Gamma = (V, U, G, N, \alpha, \beta, x_I, x_O)$$

Accepted language



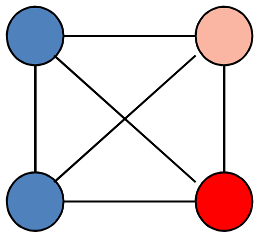
1. There exists a configuration in which the set of words existing in the output node x_O is non-empty. (halting and accepting computation)
2. There exist two consecutive identical configurations. (halting and rejection computation)
3. It works forever.

L(Γ)={$w \in V^*$: the computation of Γ on w is an accepting one}.

Networks of Bio-Inspired Processors

Accepting Networks of Splicing Processors

$$\Gamma = (V, U, G, N, \alpha, x_I, x_O)$$



Let r=u_1#u_2\$$v_1$#$v_2$ an splicing rule then $(x, y) \mapsto_r (w, z)$ iff $x=x_1u_1u_2x_2$, $y=y_1v_1v_2y_2$, $w=x_1u_1v_2y_2$ and $z=y_1v_1u_2x_2$

$$\sigma_R(L) = \{z, w \in V^* : (\exists u, v \in L, r \in R)[(u, v) \mapsto_r (z, w)]\}$$

An splicing processor over *V* is a 5-tuple *(S,A,PI,FI,PO,FO)*,where:

S is a finite set of splicing rules
A is a finite set of auxiliary words
The rest of elements are identical to the evolutionary case

Splicing steps $C'(x) = \sigma_{S_x}(C(x) \cup A_x)$

SAME ACCEPTANCE CRITERION AS IN THE EVOLUTIONARY CASE

Networks of Genetic Processors

Accepting Networks of Genetic Processors

ANGP of size *n* is a tuple $\Gamma = (V, N_1, N_2, \ldots, N_n, G, N)$

where V is an alphabet

$G=(X_G,E_G)$ is an undirected graph without loops

N_i $(1 \leq i \leq n)$ is a genetic processor over V

$N: X_G \rightarrow \{N_1,N_2,\ldots,N_n\}$ associates a genetic processor to every node in the graph

A genetic processor over *V* is a 5-tuple $(M_R,A,PI,FI,PO,FO,\alpha,\beta)$

SAME ACCEPTANCE CRITERION AS IN THE EVOLUTIONARY CASE

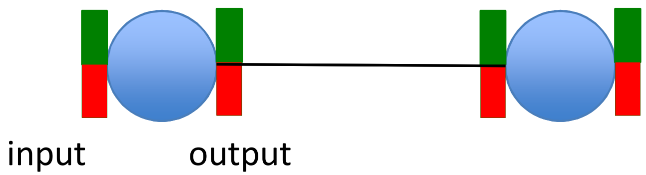
Networks of Bio-Inspired Processors

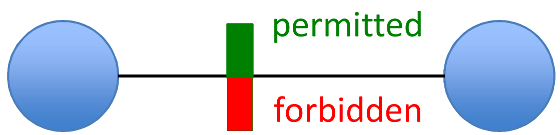
Towards a full general model …

First: Generalize the operations in the processors

Mutation processors | Splicing processors | Hairpin processors | etc. etc.

Second: Generalize the filter positions





Networks of Bio-Inspired Processors

Towards a full general model …

A bio-inspired processor over *V* is a 5-tuple *(op,PI,FI,PO,FO)*,where:

op is a biologically inspired operation over strings
PI,FI $\subseteq$ V are the input permitting/forbidding contexts of the processor

PO,FO$\subseteq$ V are the output permitting/forbidding contexts of the processor

- ***op* encapsulates the operation parameters**
- **PI,FI,PO and FO can be empty so the filters are attached to the connections**

Networks of Bio-Inspired Processors

Accepting Networks of Bio-Inspired Processors

$$\Gamma = (V, U, G, N, \beta, \gamma, x_I, x_O)$$

where

V and U are the input and network alphabets
$G=(X_G,E_G)$ is an undirected graph without loops
$N: X_G \rightarrow BP_U$ associate a bio-inspired processor to every node in G
$\beta: X_G \rightarrow \{s,w\}$ associates a filter predicate to every node
$\gamma: E_G \rightarrow 2^U \times 2^U$ associates a filter (P_e,F_e) to every edge in the graph
x_I, x_O are the input and output nodes

Networks of Bio-Inspired Processors

References

Networks of Evolutionary Processors

- J.Castellanos, Carlos Martín-Vide, Victor Mitrana, José M.Sempere. Networks of evolutionary processors. Acta Informatica 39, pp 517-529. 2003.

- M. Margenstern, V. Mitrana, M.J. Pérez-Jiménez. Accepting hybrid networks of evolutionary processors. In Proceedings of the International Meeting on DNA Computing, DNA 10, LNCS Vol. 3384, pp 235-246. Springer.2005.

Networks of Splicing Processors

- F. Manea, C. Martín-Vide, V. Mitrana. Accepting networks of splicing processors. In Proceedings of the First Conference on Computability in Europe, CiE 2005,LNCS Vol. 3526, pp 300-309. Springer. 2005.

- F. Manea, C. Martín-Vide, V. Mitrana. Accepting networks of splicing processors: complexity results. Theoretical Computer Science 371, pp 72-82. 2007.

Networks of Genetic Processors

- M. Campos, José M. Sempere. Accepting networks of genetic processors are computationally complete (*submitted*)

- M. Campos, José M. Sempere. Formal Language Generation with Networks of Genetic Processors (*submitted*)