Cell Division versus Membrane Fission:
A Computational Complexity Perspective

Mario J. Pérez Jiménez

Research Group on Natural Computing
Dpto. Ciencias de la Computación e Inteligencia Artificial
University of Sevilla
Academia Europaea (The Academy of Europe)

www.cs.us.es/˜marper marper@us.es

13th Brainstorming Week on Membrane Computing
Sevilla, Spain, February 2, 2015

Cell division (I)

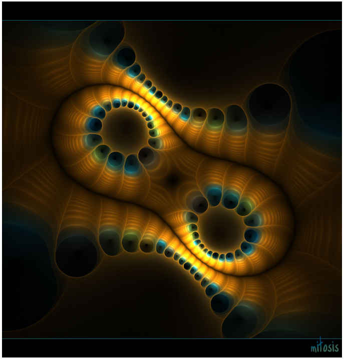
Cell division (I)

One of the basic processes in the cell life cycle.

Cell division (I)

One of the basic processes in the cell life cycle.

- *Binary fission* (prokaryotic cells)
- *Mitosis* (eukaryotic cells)
- *Meiosis* (eukaryotic cells)



Cell division (II)

[1] Gh. Păun. Computing with membranes: Attacking **NP**-complete problems. In I. Antoniou, C. Calude, M. J. Dinneen, (eds.) *Unconventional Models of Computation, UMC'2K* Springer-Verlag, 2000, pp. 94-115.

[2] C. Zandron, C. Ferreti, G. Mauri. Computing with membranes: Attacking **NP**-complete problems. In I. Antoniou, C. Calude, M. J. Dinneen, (eds.) *Unconventional Models of Computation, UMC'2K* Springer-Verlag, 2000, pp. 289-301.

[3] Gh. Păun, M.J. Pérez-Jiménez, A. Riscos-Núñez. Tissue P system with cell division. *International Journal of Computers, Communications & Control* , **Vol. III, 3** (2008), 295–303.

Cell division (II)

Cell division inspired mechanism in Membrane Computing:

- P systems with active membranes (membrane division rules)[1]
  - ⋆ Evolution, Send-in, Send-out, Dissolution, Membrane division rules.
  - ⋆ Computational completeness.
  - ⋆ Computational efficiency (a semi-uniform polynomial time solution for SAT by using communication rules with length bounded by the number of clauses of the input[2]).

[1] Gh. Păun. Computing with membranes: Attacking **NP**-complete problems. In I. Antoniou, C. Calude, M. J. Dinneen, (eds.) *Unconventional Models of Computation, UMC'2K* Springer-Verlag, 2000, pp. 94-115.

[2] C. Zandron, C. Ferreti, G. Mauri. Computing with membranes: Attacking **NP**-complete problems. In I. Antoniou, C. Calude, M. J. Dinneen, (eds.) *Unconventional Models of Computation, UMC'2K* Springer-Verlag, 2000, pp. 289-301.

[3] Gh. Păun, M.J. Pérez-Jiménez, A. Riscos-Núñez. Tissue P system with cell division. *International Journal of Computers, Communications & Control* , **Vol. III, 3** (2008), 295–303.

Cell division (II)

Cell division inspired mechanism in Membrane Computing:

- P systems with active membranes (membrane division rules)[1]
  - ⋆ Evolution, Send-in, Send-out, Dissolution, Membrane division rules.
  - ⋆ Computational completeness.
  - ⋆ Computational efficiency (a semi-uniform polynomial time solution for SAT by using communication rules with length bounded by the number of clauses of the input[2]).
- Tissue P systems with cell division (cell division rules)[3]
  - ⋆ Symport/antiport, Cell division rules.
  - ⋆ Computational completeness.
  - ⋆ Computational efficiency (a uniform polynomial time solution for SAT by using communication rules with length at most 5).

[1] Gh. Păun. Computing with membranes: Attacking **NP**-complete problems. In I. Antoniou, C. Calude, M. J. Dinneen, (eds.) *Unconventional Models of Computation, UMC'2K* Springer-Verlag, 2000, pp. 94-115.

[2] C. Zandron, C. Ferreti, G. Mauri. Computing with membranes: Attacking **NP**-complete problems. In I. Antoniou, C. Calude, M. J. Dinneen, (eds.) *Unconventional Models of Computation, UMC'2K* Springer-Verlag, 2000, pp. 289-301.

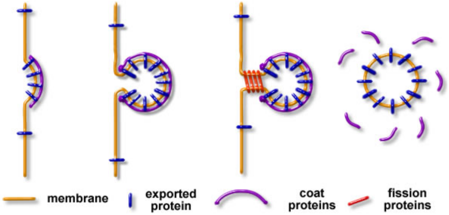
[3] Gh. Păun, M.J. Pérez-Jiménez, A. Riscos-Núñez. Tissue P system with cell division. *International Journal of Computers, Communications & Control* , **Vol. III, 3** (2008), 295–303.

Membrane fission (I)

Membrane fission (I)

Lipid membranes:

- *Plasma membrane*
  - ⋆ Separates the interior of a cell from its environment.
- *Membrane compartments*
  - ⋆ Concentrations barriers alowing incorporate material (donnor-aceptor membrane).



Membrane fission (II)

[4] A. Alhazov, T.O. Ishdorj. Membrane operations in P systems with active membranes. In Gh.Păun et al. (eds.) *Proceedings of the Second Brainstorming Week on Membrane Computing*, Sevilla, 2-7 February 2004, Research Group on Natural Computing, TR 01/2004, University of Seville, 37-44.

[5] L. Pan, T.-O. Ishdorj. P systems with active membranes and separation rules. *Journal of Universal Computer Science*, **10**, 5 (2004), 630–649.

[6] L. Pan, M.J. Pérez-Jiménez. Computational complexity of tissue–like P systems. *Journal of Complexity*, **26**, 3 (2010), 296–315.

Membrane fission (II)

Membrane fission inspired mechanism in Membrane Computing:

- P systems with active membranes:
  - Membrane separation rules associated with subsets of the working alphabet[4] (a semi-uniform polynomial time solution to SAT by using evolution rules with length at most 5).
  - Membrane separation rules associated with a prefixed partition of the working alphabet[5] (computational completeness + a uniform polynomial time solution to SAT by using evolution rules with length at most 5).
- Tissue P systems with cell separation:
  - Cell separation rules associated with a prefixed partition of the working alphabet[6] (a semi-uniform polynomial time solution to SAT by using evolution rules with length at most 8).

[4] A. Alhazov, T.O. Ishdorj. Membrane operations in P systems with active membranes. In Gh.Păun et al. (eds.) *Proceedings of the Second Brainstorming Week on Membrane Computing*, Sevilla, 2-7 February 2004, Research Group on Natural Computing, TR 01/2004, University of Seville, 37-44.

[5] L. Pan, T.-O. Ishdorj. P systems with active membranes and separation rules. *Journal of Universal Computer Science*, **10**, 5 (2004), 630–649.

[6] L. Pan, M.J. Pérez-Jiménez. Computational complexity of tissue–like P systems. *Journal of Complexity*, **26**, 3 (2010), 296–315.

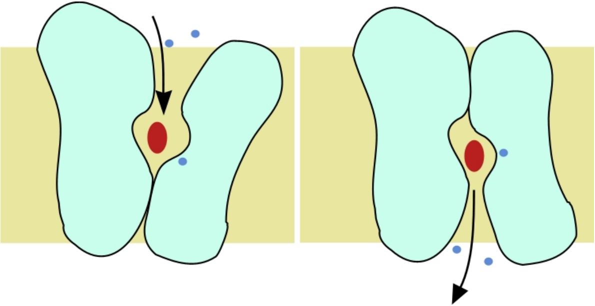
Trans-membrane transport

[7] A. Păun, Gh. Păun, G. Rozenberg. Computing by communication in networks of membranes, *International Journal of Foundations of Computer Science*, **13**, 6 (2002), 779–798.

Trans-membrane transport

Networks of membranes which compute by communication only:

- Symport/antiport rules[7] .
- Used both for communication with the environment and for direct communication between membranes.
- The environment plays an active role.
- Computational completeness.



[7] A. Păun, Gh. Păun, G. Rozenberg. Computing by communication in networks of membranes, *International Journal of Foundations of Computer Science*, **13**, 6 (2002), 779–798.

Cell-like P systems with symport/antiport rules

[8] A. Păun, Gh. Păun. The power of communication: P systems with symport/antiport, *New Generation Computing*, **20**, 3 (2002), 295–305.

Cell-like P systems with symport/antiport rules

$\Pi = (\Gamma, \mathcal{E}, \Sigma, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}_1, \cdots, \mathcal{R}_q, i_{in}, i_{out})$ [8]

- ⋆ Computational completeness.

[8] A. Păun, Gh. Păun. The power of communication: P systems with symport/antiport, *New Generation Computing*, **20**, 3 (2002), 295–305.

Cell-like P systems with symport/antiport rules

$\Pi = (\Gamma, \mathcal{E}, \Sigma, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}_1, \cdots, \mathcal{R}_q, i_{in}, i_{out})$ [8]

- ⋆ Computational completeness.

Cell-like versus Tissue-like: Symport/antiport rules:

	Cell-like	tissue-like
Set of rules	Each membrane has associated a set of rules	Associated with the system
Structure	Rooted tree: defined in an explicit way	Directed graph: defined by the set of rules
Environment	Only skin membrane can communicate with it	Any cell can communicate with it
Communication	Two membranes: in an indirect way	Two cells: directly

[8] A. Păun, Gh. Păun. The power of communication: P systems with symport/antiport, *New Generation Computing*, **20**, 3 (2002), 295–305.

For each $k \geq 1$, we consider the following classes of recognizer cell-like P systems (**set of rules associated with each membrane**)

$\boxed{\mathcal{CDC}(k)}$

- $(u, out; v, in)$, for $u, v \in \Gamma^*$ (*symport-antiport rules*) whose length $(|u| + |v|)$ is at most k.
- $[\ a\ ]_i \rightarrow [\ b\ ]_i[\ c\ ]_i$, where $i \in \{1, 2, \ldots, q\}$ and $a, b, c \in \Gamma$ (*division rules*).

For each $k \geq 1$, we consider the following classes of recognizer cell-like P systems (**set of rules associated with each membrane**)

$\boxed{\mathcal{CDC}(k)}$

- ⋆ $(u, out; v, in)$, for $u, v \in \Gamma^*$ (*symport-antiport rules*) whose length $(|u| + |v|)$ is at most k.
- ⋆ $[\ a\ ]_i \rightarrow [\ b\ ]_i[\ c\ ]_i$, where $i \in \{1, 2, \ldots, q\}$ and $a, b, c \in \Gamma$ (*division rules*).

$\boxed{\mathcal{CSC}(k)}$

- ⋆ $(u, out; v, in)$, for $u, v \in \Gamma^*$ (*symport-antiport rules*) whose length $(|u| + |v|)$ is at most k.
- ⋆ $[\ a\ ]_i \rightarrow [\ \Gamma_1\ ]_i\ [\ \Gamma_2\ ]_i$, where $i \in \{1, 2, \ldots, q\}$, $a \in \Gamma$, $i \neq i_{out}$ and $\{\Gamma_1, \Gamma_2\}$ is a fixed partition of Γ (*separation rules*).

For each $k \geq 1$, we consider the following classes of recognizer cell-like P systems (**set of rules associated with each membrane**)

$\boxed{\mathcal{CDC}(k)}$

- $(u, out; v, in)$, for $u, v \in \Gamma^*$ (*symport-antiport rules*) whose length $(|u| + |v|)$ is at most k.
- $[\ a\ ]_i \rightarrow [\ b\ ]_i [\ c\ ]_i$, where $i \in \{1, 2, \ldots, q\}$ and $a, b, c \in \Gamma$ (*division rules*).

$\boxed{\mathcal{CSC}(k)}$

- $(u, out; v, in)$, for $u, v \in \Gamma^*$ (*symport-antiport rules*) whose length $(|u| + |v|)$ is at most k.
- $[\ a\ ]_i \rightarrow [\ \Gamma_1\ ]_i\ [\ \Gamma_2\ ]_i$, where $i \in \{1, 2, \ldots, q\}$, $a \in \Gamma$, $i \neq i_{out}$ and $\{\Gamma_1, \Gamma_2\}$ is a fixed partition of Γ (*separation rules*).

The semantics is similar to the tissue P systems with cell division or separation.

Cell-like P systems with symport/antiport rules and "without environment"

$\Pi = (\Gamma, \mathcal{E}, \Sigma, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}_1, \cdots, \mathcal{R}_q, i_{in}, i_{out})$ such that $\mathcal{E} = \emptyset$.

- ⋆ No objects initially located in the environment of the system available in an arbitrary number of copies.
- ⋆ In such P systems objects in the environment always have finite multiplicity.

Cell-like P systems with symport/antiport rules and “without environment”

$\Pi = (\Gamma, \mathcal{E}, \Sigma, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}_1, \cdots, \mathcal{R}_q, i_{in}, i_{out})$ such that $\mathcal{E} = \emptyset$.

- ⋆ No objects initially located in the environment of the system available in an arbitrary number of copies.
- ⋆ In such P systems objects in the environment always have finite multiplicity.

The classes $\widehat{\mathcal{CDC}(k)}$ and $\widehat{\mathcal{CSC}(k)}$

Recent results

- $\mathbf{P} = \mathbf{PMC}_{\mathcal{CDC}(1)} = \mathbf{PMC}_{\mathcal{CSC}(1)}$ (dependency graph technique).

Recent results

- $\mathbf{P} = \mathbf{PMC}_{\mathcal{CDC}(1)} = \mathbf{PMC}_{\mathcal{CSC}(1)}$ (dependency graph technique).
- $\mathbf{P} = \mathbf{PMC}_{\mathcal{CSC}(2)}$ (algorithmic technique).

Recent results

- $\mathbf{P} = \mathbf{PMC}_{\mathcal{CDC}(1)} = \mathbf{PMC}_{\mathcal{CSC}(1)}$ (dependency graph technique).
- $\mathbf{P} = \mathbf{PMC}_{\mathcal{CSC}(2)}$ (algorithmic technique).
- $\mathbf{NP} \cup \textbf{co-NP} \subseteq \mathbf{PMC}_{\mathcal{CSC}(3)}$ ($\mathbf{SAT} \in \mathbf{PMC}_{\mathcal{CSC}(3)}$).

Recent results

- $\mathbf{P} = \mathbf{PMC}_{\mathcal{CDC}(1)} = \mathbf{PMC}_{\mathcal{CSC}(1)}$ (dependency graph technique).
- $\mathbf{P} = \mathbf{PMC}_{\mathcal{CSC}(2)}$ (algorithmic technique).
- $\mathbf{NP} \cup \textbf{co-NP} \subseteq \mathbf{PMC}_{\mathcal{CSC}(3)}$ $(\mathbf{SAT} \in \mathbf{PMC}_{\mathcal{CSC}(3)})$.
- $\mathbf{NP} \cup \textbf{co-NP} \subseteq \mathbf{PMC}_{\mathcal{CDC}(2)}$ $(\mathbf{HAM-CYCLE} \in \mathbf{PMC}_{\mathcal{CDC}(2)})$.

Recent results

- $\mathbf{P} = \mathbf{PMC}_{\mathcal{CDC}(1)} = \mathbf{PMC}_{\mathcal{CSC}(1)}$ (dependency graph technique).
- $\mathbf{P} = \mathbf{PMC}_{\mathcal{CSC}(2)}$ (algorithmic technique).
- $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{CSC}(3)}$ ($\mathbf{SAT} \in \mathbf{PMC}_{\mathcal{CSC}(3)}$).
- $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{CDC}(2)}$ ($\mathbf{HAM} - \mathbf{CYCLE} \in \mathbf{PMC}_{\mathcal{CDC}(2)}$).
- $\mathbf{P} = \mathbf{PMC}_{\widehat{\mathcal{CSC}}}$ (algorithmic technique).

Recent results

- $\mathbf{P} = \mathbf{PMC}_{\mathcal{CDC}(1)} = \mathbf{PMC}_{\mathcal{CSC}(1)}$ (dependency graph technique).
- $\mathbf{P} = \mathbf{PMC}_{\mathcal{CSC}(2)}$ (algorithmic technique).
- $\mathbf{NP} \cup \textbf{co-NP} \subseteq \mathbf{PMC}_{\mathcal{CSC}(3)}$ ($\mathbf{SAT} \in \mathbf{PMC}_{\mathcal{CSC}(3)}$).
- $\mathbf{NP} \cup \textbf{co-NP} \subseteq \mathbf{PMC}_{\mathcal{CDC}(2)}$ ($\mathbf{HAM} - \mathbf{CYCLE} \in \mathbf{PMC}_{\mathcal{CDC}(2)}$).
- $\mathbf{P} = \mathbf{PMC}_{\widehat{\mathcal{CSC}}}$ (algorithmic technique).
- For each $k \geq 1$, $\mathbf{PMC}_{\mathcal{CDC}(k)} = \mathbf{PMC}_{\widehat{\mathcal{CDC}(k)}}$ (simulation technique).

Proof techniques

Dependency graph

- Construction of a directed graph (*dependency graph*) G_Π associated with a P system Π verifying:
  - There exists an accepting computation of Π if and only if there exists a path between two distinguished nodes in the dependency graph associated with it.

Proof techniques

Dependency graph

- ⋆ Construction of a directed graph (*dependency graph*) G_Π associated with a P system Π verifying:
  - There exists an accepting computation of Π if and only if there exists a path between two distinguished nodes in the dependency graph associated with it.

Algorithmic

- ⋆ A deterministic algorithm $\mathcal{A}$ working in polynomial time that receives as input a P system Π and an input multiset m of Π. Then, algorithm $\mathcal{A}$ reproduces the behaviour of **a single** computation of $\Pi + m$.

Proof techniques

Dependency graph

- ⋆ Construction of a directed graph (*dependency graph*) G_Π associated with a P system Π verifying:
  - There exists an accepting computation of Π if and only if there exists a path between two distinguished nodes in the dependency graph associated with it.

Algorithmic

- ⋆ A deterministic algorithm $\mathcal{A}$ working in polynomial time that receives as input a P system Π and an input multiset m of Π. Then, algorithm $\mathcal{A}$ reproduces the behaviour of **a single** computation of $\Pi + m$.

Simulation

- ⋆ Π' *simulates* Π *in an efficient way* if the following holds:

  (a) Π' can be constructed from Π by a DTM working in polynomial time.

  (b) There exists an injective function, f, from $\textbf{Comp}(\Pi)$ onto $\textbf{Comp}(\Pi')$ such that:

  - There exists a DTM that constructs $f(\mathcal{C})$ from computation $\mathcal{C}$ in polynomial time.
  - A computation $\mathcal{C}$ is an accepting computation if and only if $f(\mathcal{C})$ is an accepting one.
  - There exists a polynomial function $p(n)$ verifying $|f(\mathcal{C})| \leq p(|\mathcal{C}|)$ for each $\mathcal{C} \in \textbf{Comp}(\Pi)$.

Frontiers of the efficiency

NonEfficiency *(Feasible)*	*Efficiency* *(Presumably Efficient)*	
$\mathcal{CDC}(1)$	$\mathcal{CDC}(2)$	(*length*)
$\mathcal{CSC}(2)$	$\mathcal{CSC}(3)$	(*length*)
$\mathcal{CSC}(2)$	$\mathcal{CDC}(2)$	(*kind*)
$\widehat{\mathcal{CSC}(2)}$	$\widehat{\mathcal{CDC}(2)}$	(*kind*)
$\widehat{\mathcal{CSC}}$	$\mathcal{CSC}$	(*environment*)

Frontiers of the efficiency

NonEfficiency *(Feasible)*	*Efficiency* *(Presumably Efficient)*	
$\mathcal{CDC}(1)$	$\mathcal{CDC}(2)$	*(length)*
$\mathcal{CSC}(2)$	$\mathcal{CSC}(3)$	*(length)*
$\mathcal{CSC}(2)$	$\mathcal{CDC}(2)$	*(kind)*
$\widehat{\mathcal{CSC}(2)}$	$\widehat{\mathcal{CDC}(2)}$	*(kind)*
$\widehat{\mathcal{CSC}}$	$\mathcal{CSC}$	*(environment)*

Each such frontier provides a new way to tackle the P versus NP problem.

Does structure matter?

Similar results for:

- Tissue P systems with symport/antiport rules (cell division/cell separation).
- Cell-like P systems with symport/antiport rules (cell division/cell separation).

Does structure matter?

Similar results for:

- Tissue P systems with symport/antiport rules (cell division/cell separation).
- Cell-like P systems with symport/antiport rules (cell division/cell separation).

Open questions:

1. **Cell-like** P systems with symport/antiport rules, can be efficiently simulated by **tissue** P systems with symport/antiport rules?
2. **Tissue** P systems with symport/antiport rules, can be efficiently simulated by **cell-like** P systems with symport/antiport rules?

Does structure matter?

Similar results for:

- Tissue P systems with symport/antiport rules (cell division/cell separation).
- Cell-like P systems with symport/antiport rules (cell division/cell separation).

Open questions:

1. **Cell-like** P systems with symport/antiport rules, can be efficiently simulated by **tissue** P systems with symport/antiport rules?
2. **Tissue** P systems with symport/antiport rules, can be efficiently simulated by **cell-like** P systems with symport/antiport rules?

Idea: Complexity aspects on **Tissue P systems with active cells**.