Tutorials, Surveys

Cooperative P Systems and the P Versus NP Problem

Luis Valencia-Cabrera, Miguel Á. Martínez-del-Amor, D. Orellana-Martín, Ignacio Pérez-Hurtado, Mario J. Pérez-Jiménez

Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
Universidad de Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
E-mail: {lvalencia, mdelamor, dorellana, perezh, marper}@us.es

Summary. The **P** versus **NP** problem is undoubtedly the most important open question in computer science. Frontiers of tractability or efficiency expressed by means of syntactic or semantic ingredients in the framework of *Membrane Computing*, an unconventional computing paradigm, can bring a new approach to tackle **P** versus **NP**. In this context, the role of the cooperation of objects to trigger rewriting rules is analysed in order to obtain this kind of borderlines. Besides, a relationship among cooperative rewriting rules and instances of 2–SAT problem and 3–SAT problem is highlighted and their connections with results of computational complexity theory are described.

Key words: Membrane Computing, P systems with active membranes, cooperative rules, the **P** versus **NP** problem, SAT problem.

1 Introduction

The relevance of the $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ question is not only the inherent pleasure of solving a mathematical problem, since an answer to it could dramatically affect our everyday lives. On the one hand, a negative answer to this question would confirm that the majority of current cryptographic systems are secure from a practical point of view. On the other hand, a positive answer would not only show the uncertainty about the secureness of these systems, but also this kind of answer is expected to come together with a general procedure such that it will provide a deterministic algorithm solving any **NP**-complete problem in polynomial time. In an informal

way, we can say that if $\mathbf{P} = \mathbf{NP}$ then it would be possible to find solutions to search problems as easily as checking whether those solutions are "correct", that is, almost all the algorithmic challenges that we face today could be solved in a "practical way" and computers could solve almost any "mechanical task". The search of techniques other than the classical ones that allow us to tackle this problem becomes a very important challenge. In this context, frontiers of tractability expressed by means of syntactic or semantic ingredients in the framework of *Membrane Computing* can bring a new approach.

Membrane computing is a flexible and versatile branch of natural computing, which arises as an abstraction of the compartmentalized structure of living cells, and the way biochemical substances are processed in (or moved between) membrane bounded regions [21]. Inspired by the structure of living cells, two main classes of membrane systems have been investigated: a hierarchical (cell-like) arrangement of membranes, inspired from the structure of the cell [21] and a net of membranes (placed in the nodes of a directed graph), inspired from the cell-interconnection in tissues [16] or inspired from the way that neurons communicate with each other by means of short electrical impulses (spikes), emitted at precise moments of time [10]. All classes of computing devices considered in the field of membrane computing are generally called *P systems* or *membrane systems*, which are parallel and distributed computational models based on processing multisets of objects in cell-like or tissue-like structures by means of rewriting rules. A P system is *cooperative* if it contains rules using cooperation, that is, rules that need more than one object to be triggered.

This paper is devoted to study the role of cooperation of objects to trigger rewriting rules in order to obtain frontiers of efficiency by means of ingredients of membrane systems working in cell-like mode. For that, the use of cooperation in rewriting rules is combined with some syntactical ingredients and the behaviour of different "cocktails" are analysed. The paper is structured as follows. In next section, some general concepts are briefly described in order to make the work self-contained. Section 3 is devoted to present the different models of cell-like membrane systems that will be studied in this work, emphasizing the syntactic and semantic aspects of them. In the following section, frontiers of efficiency involving cooperation are presented in the different models considered through the paper. In Section 5, a relationship among cooperative rewriting rules and instances of 2–`SAT` and 3–`SAT` is highlighted from a complexity point of view. Finally, some conclusions are discussed.

2 Preliminaries

An *alphabet* Γ is a non-empty set and their elements are called *symbols*. A *string* u over Γ is an ordered finite sequence of symbols, that is, a mapping from a natural number $n \in \mathbb{N}$ onto Γ. The number n is called the *length* of the string u and it is

denoted by $|u|$. The empty string (with length 0) is denoted by λ. The set of all strings over an alphabet Γ is denoted by Γ^*. A *language* over Γ is a subset of Γ^*.

A *multiset* over an alphabet Γ is an ordered pair (Γ, f) where f is a mapping from Γ onto the set of natural numbers $\mathbb{N}$. The *support* of a multiset $m = (\Gamma, f)$ is defined as $supp(m) = \{x \in \Gamma \mid f(x) > 0\}$. A multiset is finite (resp. empty) if its support is a finite (resp. empty) set. We denote by $\emptyset$ the empty multiset and we denote by $M(\Gamma)$ the set of all multisets over Γ.

Let $m_1 = (\Gamma, f_1)$, $m_2 = (\Gamma, f_2)$ be multisets over Γ, then the union of m_1 and m_2, denoted by $m_1 + m_2$, is the multiset (Γ, g), where $g(x) = f_1(x) + f_2(x)$ for each $x \in \Gamma$.

2.1 Graphs and trees

Let us recall some notions related with graph theory (see [6] for details). An *undirected graph* is an ordered pair (V, E) where V is a set whose elements are called nodes or vertices and $E = \{\{x, y\} \mid x \in V, y \in V, x \neq y\}$ whose elements are called *edges*. A *directed graph* is an ordered pair (V, E) where V is a set whose elements are called nodes or vertices and $E = \{(x, y) \mid x \in V, y \in V\}$ whose elements are called *arcs*. A *path* of length $k \geq 1$ from a node u to a node v in a graph (V, E) is a finite sequence $(x_0, x_1, \ldots, x_k)$ of nodes such that $x_0 = u$, $x_k = v$ and $(x_i, x_{i+1}) \in E$ for $0 \leq i \leq k-1$ (in the case of a directed graph or $\{x_i, x_{i+1}\} \in E$ in the case of an undirected graph. If $k \geq 2$ and $x_0 = x_k$ then we say that the path is a *cycle* of the graph. A graph with no cycle is said to be *acyclic*. An undirected graph is *connected* if there exist paths between every pair of nodes.

A *free tree* (*tree*, for short) is a connected, acyclic, undirected graph. A *rooted tree* is a tree in which one of the vertices (called *the root of the tree*) is distinguished from the others. In a rooted tree the concepts of ascendants and descendants are defined in a usual way. Given a node x (different from the root), if the last edge on the (unique) path from the root of the tree to the node x is $\{x, y\}$ (in this case, $x \neq y$), then y is **the** *parent* of node x and x is **a** *child* of node y. The root is the only node in the tree with no parent. A node with no children is called a *leaf*.

2.2 Decision problems

Roughly speaking, a decision problem X is one whose solution/answer is either "`yes`" or "`no`". This can be formally defined by an ordered pair (I_X, θ_X), where I_X is a language over a finite alphabet Σ_X and θ_X is a total boolean function over I_X. The elements of I_X are called *instances* of the problem X. Each decision problem X has associated a language L_X over the alphabet Σ_X as follows: $L_X = \{u \in E_X \mid \theta_X(u) = 1\}$. Conversely, every language L over an alphabet Σ has associated a decision problem $X_L = (I_{X_L}, \theta_{X_L})$ as follows: $I_{X_L} = \Sigma^*$ and $\theta_{X_L}(u) = 1$ if and only if $u \in L$. Then, given a decision problem X we have $X_{L_X} = X$, and given a language L over an alphabet Σ we have $L_{X_L} = L$.

It is worth pointing out that any Turing machine M (with input alphabet Σ_M) has associated a *decision problem* $X_M = (I_M, \theta_M)$ defined as follows: $I_M = \Sigma_M^*$, and for every $w \in \Sigma_M^*$, $\theta_M(w) = 1$ if and only if M accepts w. Obviously, the decision problem X_M is solvable by the Turing machine M.

The satisfiability problem

The satisfiability problem (`SAT` problem) is described as follows: *given a boolean formula in conjunctive normal form (CNF), to determine whether or not there exists an assignment to its variables, called truth assignment, on which it evaluates true*. `SAT` is the first problem which was demonstrated that belongs to the class of **NP**-complete problems [7]. These problems are in the class **NP** and they verify an interesting property: its individual complexity can be extended to the entire class **NP**, that is, if `SAT` is a problem in **P** then all problems in **NP** also belongs to **P**. Different variants of the `SAT` problem were considered, in particular, for each $k \geq 1$, k–`SAT` problem is a special case of the `SAT` problem in which all clauses of the input formula have exactly k literals. It is well known that 2–`SAT` is in class **P** (in fact, it is an **NL**) and 3–`SAT` is an **NP**-complete problem [7].

The reachability problem

The reachability problem is described as follows: *given a directed graph $G = (V, E)$ with two specified vertices s and t, determine whether or not there is a path from s to t*. There are algorithms solving this problem, for instance, search algorithms based on breadth-first search or depth-first search. These algorithms determine whether two vertices are connected in $O(\max(|V|, |E|))$ time. Moreover, they basically need to store at most $|V|$ items, so these algorithms use $O(|V|)$ space. But this quantity of space can be reduced to $O(\log^2|V|)$ by using an algorithm that could be called middle-first search (see [20] for details, pp. 149-150). In particular, reachability problem is in class **P**.

3 Cell-like membrane systems

In order to make this paper self-contained, the different models of membrane systems considered in this work are introduced detailing their syntax and semantics.

3.1 Basic transition P systems

Next, the basic model introduced by Gh. Păun in its seminal paper [21] is presented by using a slightly different notation.

Definition 1. *A* basic transition P system *of degree $q \geq 1$ is a tuple $\Pi = (\Gamma, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, \mathcal{P}_\mathcal{R}, i_{out})$, where:*

1. Γ *is a finite alphabet.*
2. μ *is a rooted tree.*
3. $\mathcal{M}_1, \ldots, \mathcal{M}_q$ *are multisets over* Γ.
4. $\mathcal{R}$ *is a finite set of rules of the following forms:*
   a) $[\, u \,]_i \rightarrow v_1 \, [\, v_2 \, [\, v_3 \,]_j \,]_i$, *for* $i, j \in \{1, \ldots, q\}$, $i \neq j$ *and* $u, v_1, v_2, v_3 \in M(\Gamma)$.
   b) $[\, u \,]_i \rightarrow v_1 \, [\, v_2 \, [\, v_3 \,]_j \,]_i \, \delta$, *for* $i, j \in \{1, \ldots, q\}$, $i \neq j$ *and* $u, v_1, v_2, v_3 \in M(\Gamma)$, *and* δ *is a distinguished symbol such that* $\delta \notin \Gamma$.
5. $\mathcal{P}_{\mathcal{R}} \subseteq \mathcal{R} \times \mathcal{R}$ *is an strict partial order over* $\mathcal{R}$.
6. $i_{out} \in \{0, 1, \ldots, q\}$.

A *basic transition P system* $\Pi = (\Gamma, \mu, , \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, \mathcal{P}_{\mathcal{R}}, i_{out})$, of degree $q \geq 1$ can be viewed as a set of q membranes injectively labelled by $1, \ldots, q$, arranged in a hierarchical structure μ given by a rooted tree whose root is called the *skin membrane* of the system and with an environment labelled by 0 such that: (a) $\mathcal{M}_1, \ldots, \mathcal{M}_q$ are multisets over the *working alphabet* Γ representing the objects initially placed in the q membranes of the system; (b) $\mathcal{R}$ is the set of rules that allows to evolve the system; (c) $\mathcal{P}_{\mathcal{R}} \subseteq \mathcal{R} \times \mathcal{R}$ is a strict partial order relation over $\mathcal{R}$ providing priorities between rules, in such a manner that if $(r_1, r_2) \in \mathcal{P}_{\mathcal{R}}$ we say that rule r_1 has a higher priority than r_2 and we denote it by $r_1 > r_2$; and (c) $i_{out} \in \{0, 1, 2, \ldots, q\}$ represents a distinguished *zone* which will encode the output of the system. We use the term *zone* i, $0 \leq i \leq q$, to refer to membrane i in the case $1 \leq i \leq q$ and to refer to the environment in the case $i = 0$.

An *instantaneous description* or a *configuration* at an instant t of a basic transition P system $\Pi = (\Gamma, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, \mathcal{P}_{\mathcal{R}}, i_{out})$ is described by the membrane structure at instant t and all multisets of objects over Γ associated with all the membranes present in the system. The *initial configuration* of the system is $(\mu, \mathcal{M}_1, \cdots, \mathcal{M}_q)$.

A rule $r \equiv [\, u \,]_i \rightarrow v_1 \, [\, v_2 \, [\, v_3 \,]_j \,]_i$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if the following holds: (a) membrane i is in $\mathcal{C}_t$; (b) multiset u is contained in such membrane; (c) j is the label of a membrane immediately inside membrane i; and (d) there is no a rule r' associated with membrane i applicable to $\mathcal{C}_t$ such that r' has a higher priority than r, that is, $(r', r) \in \mathcal{P}_{\mathcal{R}}$. When applying such a rule, the objects specified by multiset u are consumed (multiset u is substracted from the multiset of membrane i), the objects specified by multiset v_1 will be moved to the zone immediately outside membrane i, the parent $p(i)$ of that membrane (this zone is the environment in the case when i is the skin membrane: in this case, the objects leave the system and they never come back), the objects specified by multiset v_2 will be placed in the same membrane i, and the objects specified by multiset v_3 in the membrane with label j which must be a membrane immediately inside membrane i.

A rule $[\, u \,]_i \rightarrow v_1 \, [\, v_2 \, [\, v_3 \,]_j \,]_i \, \delta$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if the following holds: (a) the rule $[\, u \,]_i \rightarrow v_1 \, [\, v_2 \, [\, v_3 \,]_j \,]_i$ is applicable to $\mathcal{C}_t$; (b) $i \neq i_{out}$; and (c) i is not the label of the skin membrane. When applying the rule $[\, u \,]_i \rightarrow v_1 \, [\, v_2 \, [\, v_3 \,]_j \,]_i \, \delta$ to a configuration $\mathcal{C}_t$, first the rule $[\, u \,]_i \rightarrow v_1 \, [\, v_2 \, [\, v_3 \,]_j \,]_i$ is

applied to $\mathcal{C}_t$ and then membrane i is dissolved. After dissolving a membrane, all objects and membranes previously present in it become elements of the contents of the immediately upper membrane which has not been dissolved.

Given a basic transition P system Π, we say that configuration $\mathcal{C}_t$ yields configuration $\mathcal{C}_{t+1}$ in one *transition step*, denoted by $\mathcal{C}_t \Rightarrow_\Pi \mathcal{C}_{t+1}$, if we can pass from $\mathcal{C}_t$ to $\mathcal{C}_{t+1}$ by applying the rules from $\mathcal{R}$ synchronously, in a *non-deterministic maximally parallel manner*. This means the following: the objects to evolve in a transition step and the rules by which they evolve are chosen in a non-deterministic manner, but in such a way that in each membrane we have a maximally parallel application of rule (at each transition step a multiset of rules which is maximal is applied, no further applicable rule can be added). A *computation* of Π is a (finite or infinite) sequence of configurations such that: (a) the first term of the sequence is the initial configuration of the system; (b) each non-first term of the sequence is obtained from the previous configuration by applying rules of the system in a non-deterministic maximally parallel manner; and (c) if the sequence is finite (called *halting computation*) then the last term of the sequence is a *halting configuration*, that is, a configuration where no rule of the system is applicable to it.

All computations start from an initial configuration and proceed as stated above; only halting computations give a result, which is encoded by the objects present in the output region i_{out} in the halting configuration.

Basic transition P systems have the ability to create an exponential workspace, expressed in terms of number of objects, in linear time (e.g. via evolution rules of the type $[\, a \rightarrow a^2 \,]_h$)

3.2 P systems with active membranes and electrical charges

Cell-like P systems with active membranes having associated electrical charges with membranes were first introduced by Gh. Păun [22]. One of the main attractive of these membrane systems is the ability to create an exponential workspace, expressed in terms of number of objects and number of membranes, in linear time.

Definition 2. *A P system with active membranes of degree $q \geq 1$ is a tuple $\Pi = (\Gamma, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$, where:*

1. *Γ is a finite alphabet.*
2. *μ is a rooted tree.*
3. *$\mathcal{M}_1, \ldots, \mathcal{M}_q$ are multisets over Γ.*
4. *$\mathcal{R}$ is a finite set of rules (H denotes the set $\{1, \ldots, q\}$) of the following forms:*
   a) *$[\, a \rightarrow u \,]_h^\alpha$, for $h \in H$,$\alpha \in \{+, -, 0\}$, $a \in \Gamma$, $u \in \Gamma^*$* (object-evolution rules).
   b) *$a\,[\ ]_h^{\alpha_1} \rightarrow [\, b \,]_h^{\alpha_2}$, for $h \in H$, $\alpha_1, \alpha_2 \in \{+, -, 0\}$, $a, b \in \Gamma$* (send–in communication rules).
   c) *$[\, a \,]_h^{\alpha_1} \rightarrow [\ ]_h^{\alpha_2}\, b$, for $h \in H$, $\alpha_1, \alpha_2 \in \{+, -, 0\}$, $a, b \in \Gamma$* (send–out communication rules).

d) $[\,a\,]_h^\alpha \to b$, *for* $h \in H$, $\alpha \in \{+,-,0\}$, $a,b \in \Gamma$ (dissolution rules).
e) $[\,a\,]_h^{\alpha_1} \to [\,b\,]_h^{\alpha_2}\,[\,c\,]_h^{\alpha_3}$, *for* $h \in H$, $\alpha_1,\alpha_2,\alpha_3 \in \{+,-,0\}$, $a,b,c \in \Gamma$ (division rules for elementary membranes).
f) $[\,[\,]_{h_1}^{\alpha_1} \ldots [\,]_{h_k}^{\alpha_1}\,[\,]_{h_{k+1}}^{\alpha_2} \ldots [\,]_{h_n}^{\alpha_2}\,]_h^{\alpha} \to [\,[\,]_{h_1}^{\alpha_3} \ldots [\,]_{h_k}^{\alpha_3}\,]_h^{\beta}\,[\,[\,]_{h_{k+1}}^{\alpha_4} \ldots [\,]_{h_n}^{\alpha_4}\,]_h^{\gamma}$, *for* $k \geq 1$, $n > k$, $h,h_1,\ldots,h_n \in H$, $\alpha,\beta,\gamma,\alpha_1,\ldots,\alpha_4 \in \{+,-,0\}$ *and* $\{\alpha_1,\alpha_2\} = \{+,-\}$ (division rules for non–elementary membranes).

5. $i_{out} \in H \cup \{0\}$.

A *P system with active membrane* $\Pi = (\Gamma,\mu,,\mathcal{M}_1,\ldots,\mathcal{M}_q,\mathcal{R},i_{out})$, of degree $q \geq 1$ can be viewed as a set of q membranes injectively labelled by $1,\ldots,q$ with electrical charges (positive $+$, negative $-$ or neutral 0) associated to them, arranged in a hierarchical structure μ given by a rooted tree whose root is called the *skin membrane* of the system and with an environment labelled by 0 such that: (a) $\mathcal{M}_1,\ldots,\mathcal{M}_q$ are multisets over the *working alphabet* Γ representing the objects initially placed in the q membranes of the system; (b) $\mathcal{R}$ is the set of rules that allows to evolve the system; and (c) $i_{out} \in \{0,1,2,\ldots,q\}$ represents a distinguished *zone* which will encode the output of the system (the term *zone* is used as above).

Notice that P systems with active membranes have some important features: (a) they use three electrical charges; (b) the polarization of a membrane, but not the label, can be modified by the application of a rule; (c) the rules are non-cooperative (the corresponding left-hand side consist of only one symbol), and (d) there are no priorities among rules.

The concept of *configuration* at an instant t of a P system with active membranes $\Pi = (\Gamma,\mu,\mathcal{M}_1,\ldots,\mathcal{M}_q,\mathcal{R},i_{out})$ is defined in a similar way to the one used in basic transition P systems.

An object-evolution rule $[\,a \to u\,]_h^\alpha$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if the following holds: (a) a membrane labelled by h is in $\mathcal{C}_t$ and its electrical charge is α; and (b) object a is contained in such membrane. When applying such a rule, object a is consumed and the objects specified by multiset u will be placed in that membrane h.

A send-in rule $a\,[\;\,]_h^{\alpha_1} \to [\,b\,]_h^{\alpha_2}$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if the following holds: (a) a membrane labelled by h, different from the skin membrane, is in $\mathcal{C}_t$ and its electrical charge is α_1; and (b) object a is contained in the zone immediately outside the membrane h, the parent $p(h)$ of that membrane. When applying such a rule, object a is consumed, object b will be placed in the same membrane h and the polarization of such membrane h will change to α_2.

A send-out rule $[\;a\;]_h^{\alpha_1} \to b\,[\;]_h^{\alpha_2}$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if the following holds: (a) a membrane labelled by h, different from the skin membrane, is in $\mathcal{C}_t$ and its electrical charge is α_1; and (b) object a is contained in such membrane h. When applying such a rule, object a is consumed, object b will be placed in the zone immediately outside the membrane h, the parent $p(h)$ of that membrane, and the polarization of such membrane h will change to α_2.

A dissolution rule $[\,a\,]_h^\alpha \to b$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if the following holds: (a) a membrane labelled by h is in $\mathcal{C}_t$ and its electrical charge is α; and (b) object a is contained in such membrane h. When applying such a rule, object a is consumed and the membrane is dissolved. After dissolving such membrane h, all objects and membranes previously present in it become elements of the contents of the immediately upper membrane which has not been dissolved, except object a triggering the rule that evolves to b.

A division rule for elementary membranes $[\,a\,]_h^{\alpha_1} \to [\,b\,]_h^{\alpha_2}\ [\,c\,]_h^{\alpha_3}$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if the following holds: (a) an elementary membrane labelled by h, different from the skin membrane, is in $\mathcal{C}_t$ and its electrical charge is α_1; (b) $h \neq i_{out}$; and (c) object a is contained in such membrane h. When applying such a rule, object a is consumed and the membrane is divided into two membranes with the same label h, maybe of different polarizations α_2 and α_3; the object a specified in the rule is replaced in the two new membranes by possibly new objects b and c, respectively.

A division rule for non-elementary membranes $[\,[\,]_{h_1}^{\alpha_1} \ldots [\,]_{h_k}^{\alpha_1}\ \ [\,]_{h_{k+1}}^{\alpha_2} \ldots [\,]_{h_n}^{\alpha_2}\,]_h^\alpha \to [\,[\,]_{h_1}^{\alpha_3} \ldots [\,]_{h_k}^{\alpha_3}\,]_h^\beta\ \ [\,[\,]_{h_{k+1}}^{\alpha_4} \ldots [\,]_{h_n}^{\alpha_4}\,]_h^\gamma$, is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if the following holds: (a) a non-elementary membrane labelled by h, different from the skin membrane, is in $\mathcal{C}_t$ and its electrical charge is α_1; (b) such membrane contains membranes with labels $h_1, \ldots, h_n$ some of them $(h_1, \ldots, h_k)$ with electrical charge α_1 and the remaining $(h_{k+1}, \ldots, h_n)$ with electrical charges α_2, being $\{\alpha_1, \alpha_2\} = \{+, -\}$; (c) if such membrane h_0 contains other membranes than those with labels $h_1, \ldots, h_n$ then they must have neutral charges. When applying such a rule: (1) membrane h_0 is divided into two membranes with the same label h, maybe of different polarizations β and γ; (2) membranes with label $h_1, \ldots, h_k$ contained in membrane h_0 are placed (with polarization α_3) inside of one of the new created membranes; (3) membranes with label $h_{k+1}, \ldots, h_n$ contained in membrane h_0 are placed (with polarization α_4) inside of the another new created membrane; and (4) if membrane h_0 contains other membranes than those with labels $h_1, \ldots, h_n$ then they have neutral charges being duplicated in the new created membranes.

P systems with active membranes differ from the basic transition P systems on the type of rules which are applied according to the following principles ([22]):

- All the rules are applied in parallel. In each transition step, one object of a membrane can be used by only one rule (chosen in a non deterministic way).
- If a membrane is dissolved, its content (multiset and internal membranes) is left free in the surrounding region.
- If at the same time a membrane labelled by h is divided by a rule of type (e)-(f) and there are objects in this membrane which evolve by means of rules of type (a), then we suppose that first the evolution rules of type (a) are used, and then the division is produced. Of course, this process takes only one step.
- The rules associated with membranes labelled by h are used for all copies of this membrane. At one step, a membrane can be the subject of *only one* rule

of types (b)-(f), that is, these rules are applied in a sequential manner to each membrane.

Given a P system with active membranes Π, we say that configuration $\mathcal{C}_t$ yields configuration $\mathcal{C}_{t+1}$ in one *transition step*, denoted by $\mathcal{C}_t \Rightarrow_\Pi \mathcal{C}_{t+1}$, if we can pass from $\mathcal{C}_t$ to $\mathcal{C}_{t+1}$ by applying the rules from $\mathcal{R}$ synchronously, in a *non-deterministic maximally parallel manner* according with the previous remarks. The concept of *computation* of Π is defined in a similar way that in the previous section.

3.3 Polarizationless P systems with active membranes

P systems with active membranes and without electrical charges were initially studied in [2, 3]. In these systems, polarizations were replaced by the possibility to change the label of the membranes by means of some rules. However, in order to obtain polynomial-time solutions to computationally hard problems, two polarizations suffice (see [4] for details). In [28] bi-stable catalysts are used to compensate the loss of computational efficiency represented by avoiding polarizations.

Definition 3. *A* polarizationless P system with active membranes *of degree* $q \geq 1$ *is a tuple* $\Pi = (\Gamma, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$, *where:*

1. Γ *is a finite alphabet.*
2. μ *is a rooted tree.*
3. $\mathcal{M}_1, \ldots, \mathcal{M}_q$ *are multisets over* Γ.
4. $\mathcal{R}$ *is a finite set of rules* (H *denotes the set* $\{1, \ldots, q\}$), *of the following forms:*
   (a) $[\, a \rightarrow u \,]_h$, *for* $h \in H$, $a \in \Gamma$, $u \in \Gamma^*$ (object-evolution rules).
   (b) $a\,[\ \ ]_h \rightarrow [\, b \,]_h$, *for* $h \in H$, $a, b \in \Gamma$ (send–in communication rules).
   (c) $[\, a \,]_h \rightarrow [\ \ ]_h\, b$, *for* $h \in H$, $a, b \in \Gamma$ (send–out communication rules).
   (d) $[\, a \,]_h \rightarrow b$, *for* $h \in H$, $a, b \in \Gamma$ (dissolution rules).
   (e) $[\, a \,]_h \rightarrow [\, b \,]_h\, [\, c \,]_h$, *for* $h \in H$, $a, b, c \in \Gamma$ (division rules for elementary *or weak* division rules for non-elementary membranes).
   (f) $[\, [\,]_{h_1} \ldots [\,]_{h_k}\, [\,]_{h_{k+1}} \ldots [\,]_{h_n}\,]_h \rightarrow [\, [\,]_{h_1} \ldots [\,]_{h_k}\,]_h\, [\, [\,]_{h_{k+1}} \ldots [\,]_{h_n}\,]_h$, *where* $k \geq 1$, $n > k$, $h, h_1, \ldots, h_n \in H$ (division rules for non-elementary membranes).
5. $i_{out} \in \{0, 1, \ldots, q\}$.

The semantics of these rules are similar to the ones of P systems with active membranes and they are applied according to usual principles of these systems, described in the previous section.

3.4 P systems with symport/antiport rules

In this section we introduce a kind of cell-like P systems that use communication rules capturing the biological phenomenon of trans-membrane transports of several chemical substances, in the same or in opposite directions. Specifically, two processes have been considered. The first one allows a multiset of chemical substances to pass through a membrane in the same direction. In the second one, two

multisets of chemical substances (located in different biological membranes) only pass with the help of each other (an *exchange* of objects between both membranes).

Next, we introduce an abstraction of these operation in the framework of P systems with symport/antiport rules following [23]. In these models, the membranes are not polarized.

Definition 4. *A P system with symport/antiport rules of degree $q \geq 1$ is a tuple $\Pi = (\Gamma, \mathcal{E}, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$, where:*

1. *Γ is a finite* alphabet.
2. *$\mathcal{E} \subseteq \Gamma$.*
3. *μ is a rooted tree.*
4. *$\mathcal{M}_1, \ldots, \mathcal{M}_q$ are multisets over Γ.*
5. *$\mathcal{R} = \mathcal{R}_1 \cup \cdots \cup \mathcal{R}_q$, where $\mathcal{R}_i$ is a finite set of rules associated with membrane i, of the following forms:*
   - ⋆ Symport rules*: (u, out) or (u, in), where $u \in M(\Gamma)$ such that $|u| > 0$;*
   - ⋆ Antiport rules*: $(u, out; v, in)$, where $u, v \in M(\Gamma)$ such that $|u| > 0$ and $|v| > 0$;*
6. *$i_{out} \in \{0, 1, \ldots, q\}$.*

A P system with symport/antiport rules of degree q

$$\Pi = (\Gamma, \mathcal{E}, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$$

can be viewed as a set of q membranes, labelled by $1, \ldots, q$, arranged in a hierarchical structure μ given by a rooted tree whose root is called the *skin membrane* of the system, labelled by 1, and with an environment labelled by 0 such that: (a) $\mathcal{M}_1, \ldots, \mathcal{M}_q$ are multisets over the *working alphabet* Γ representing the objects initially placed in the q membranes of the system; (b) $\mathcal{E}$ is the set of objects initially located in the environment of the system, all of them available in an arbitrary number of copies; (c) $\mathcal{R} = \mathcal{R}_1 \cup \cdots \cup \mathcal{R}_q$ where $\mathcal{R}_i$ is a finite set of communication rules over Γ associated with membrane i of μ; and (d) i_{out} represents a distinguished *zone* which will encode the output of the system (the term zone is used as before). The length of rule (u, out) or (u, in) (resp. $(u, out; v, in)$) is defined as $|u|$ (resp. $|u| + |v|$).

A P system with symport/antiport rules of degree $q \geq 1$

$$\Pi = (\Gamma, \mathcal{E}, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$$

where $\mathcal{E} = \emptyset$, is called a *P system with symport/antiport rules and without environment.*

For each membrane $i \in \{2, \ldots, q\}$, different from the skin membrane, we denote by $p(i)$ the parent of membrane i in the rooted tree μ. We define $p(1) = 0$, that is, by convention the "parent" of the skin membrane is the environment.

An *instantaneous description* or a *configuration* at an instant t of a P system with symport/antiport rules is described by the membrane structure at instant t,

all multisets of objects over Γ associated with all the membranes present in the system, and the multiset of objects over $\Gamma - \mathcal{E}$ associated with the environment at that moment. Recall that initially there are infinite copies of objects from $\mathcal{E}$ in the environment, and hence this set is not properly changed along the computation. The *initial configuration* of the system is $(\mu, \mathcal{M}_1, \cdots, \mathcal{M}_q; \emptyset)$.

A symport rule $(u, out) \in \mathcal{R}_i$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if the following holds: (a) a membrane labelled by i is in $\mathcal{C}_t$; and (b) multiset u is contained in such membrane. When applying a rule $(u, out) \in \mathcal{R}_i$, the objects specified by multiset u are sent out of such membrane i into the region immediately outside, the parent $p(i)$ of such membrane. This can be the environment in the case of the skin membrane.

A symport rule $(u, in) \in \mathcal{R}_i$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if the following holds:: (a) a membrane labelled by i is in $\mathcal{C}_t$; and (b) multiset u is contained in the parent $p(i)$ of such membrane i. When applying a rule $(u, in) \in \mathcal{R}_i$, the objects specified by multiset u goes out from the parent $p(i)$ membrane of i and enters into the region defined by the membrane i.

An antiport rule $(u, out; v, in) \in \mathcal{R}_i$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if if the following holds: (a) a membrane labelled by i is in $\mathcal{C}_t$; (b) multiset u is contained in such membrane; and (c) multiset v is contained in the parent $p(i)$ of membrane i. When applying a rule $(u, out; v, in) \in \mathcal{R}_i$, the objects specified by multiset u are sent out of membrane i into the parent $p(i)$ of i and, at the same time, bringing the objects specified by multiset v into such membrane i.

The rules of a P system with symport/antiport rules are applied in a non-deterministic maximally parallel manner: at each step we apply a multiset of rules which is maximal, so no further applicable rule can be added.

Given a P system with symport/antiport rules Π, we say that configuration $\mathcal{C}_t$ yields configuration $\mathcal{C}_{t+1}$ in one *transition step*, denoted by $\mathcal{C}_t \Rightarrow_\Pi \mathcal{C}_{t+1}$, if we can pass from $\mathcal{C}_t$ to $\mathcal{C}_{t+1}$ by applying the rules from $\mathcal{R}_1 \cup \cdots \cup \mathcal{R}_q$ following the previous remarks. The concept of *computation* of Π is defined in a similar way that in the previous section.

P systems with symport/antiport rules and membrane division or membrane separation

In this section, we introduce new types of rules (membrane division and membrane separation) inspired by the mitosis and the membrane fission processes, in the framework of P systems with symport/antiport rules. These rules provide a mechanism to construct an exponential workspace (expressed in terms of number of objects and number of membranes) in linear time.

Definition 5. *A* P system with symport/antiport rules and membrane division *of degree $q \geq 1$ is a tuple $\Pi = (\Gamma, \mathcal{E}, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$, where:*

1. *$\Pi = (\Gamma, \mathcal{E}, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$ is a P system with symport/antiport rules.*

2. $\mathcal{R} = \mathcal{R}_1 \cup \cdots \cup \mathcal{R}_q$, *where* $\mathcal{R}_i$ *is a finite set of symport/antiport rules, associated with membrane* i*, which can also contain rules of the following form:* $[a]_i \rightarrow [b]_i[c]_i$*, where* $i \notin \{1, i_{out}\}$ *and* $a, b, c \in \Gamma$ (division rules for elementary membranes).

A division rule $[a]_i \rightarrow [b]_i[c]_i \in \mathcal{R}_i$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if the following holds: (a) a membrane labelled by i, different from the skin membrane, is in $\mathcal{C}_t$; (b) $i \neq i_{out}$; and (c) object a is contained in such membrane. When applying a division rule $[a]_i \rightarrow [b]_i[c]_i$, under the influence of object a, the membrane with label i is divided into two membranes with the same label; in the first copy, object a is replaced by object b, in the second one, object a is replaced by object c; all the other objects residing in such membrane i are replicated and copies of them are placed in the two new membranes.

Definition 6. *A P system with symport/antiport rules and membrane separation of degree* $q \geq 1$ *is a tuple*

$$\Pi = (\Gamma, \Gamma_0, \Gamma_1, \mathcal{E}, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$$

where:

1. $\Pi = (\Gamma, \mathcal{E}, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$ *is a P system with symport/antiport rules.*
2. $\{\Gamma_0, \Gamma_1\}$ *is a partition of* Γ*, that is,* $\Gamma = \Gamma_0 \cup \Gamma_1$*,* $\Gamma_0, \Gamma_1 \neq \emptyset$*,* $\Gamma_0 \cap \Gamma_1 = \emptyset$*;*
3. $\mathcal{R} = \mathcal{R}_1 \cup \cdots \cup \mathcal{R}_q$ *where* $\mathcal{R}_i$ *is a finite set, associated with membrane* i*, of rules symport/antiport rules which can also contain rules of the following form:* $[a]_i \rightarrow [\Gamma_0]_i[\Gamma_1]_i$*, where* $i \notin \{1, i_{out}\}$ *and* $a \in \Gamma$ (separation rules).

A separation rule $[a]_i \rightarrow [\Gamma_0]_i[\Gamma_1]_i \in \mathcal{R}_i$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if the following holds: (a) a membrane labelled by i, different from the skin membrane, is in $\mathcal{C}_t$; (b) $i \neq i_{out}$; and (c) object a is contained in such membrane. When applying a separation rule $[a]_i \rightarrow [\Gamma_0]_i[\Gamma_1]_i \in \mathcal{R}_i$, in reaction with an object a, the membrane i is separated into two membranes with the same label; at the same time, object a is consumed; the objects from Γ_0 are placed in the first membrane, those from Γ_1 are placed in the second membrane.

With respect to the semantics of these variants, the rules of such P systems are applied in a non-deterministic maximally parallel manner, with the following important remark: when a membrane i is divided (resp. separated), the division rule (resp. separation rule) is the only one from $\mathcal{R}_i$ which is applied for that membrane at that step (however, some rules can be applied in a daughter membrane). The new membranes resulting from division (resp. separation) could participate in the interaction with other membranes or the environment by means of communication rules at the next step – providing that they are not divided (resp. separated) once again. The label of a membrane precisely identify the rules which can be applied to it.

3.5 Recognizer membrane systems

Let us recall that solving a decision problem can be expressed in terms of recognizing the language associated with it. Recognizer P systems were introduced in [32] and they provide a natural framework to solve decision problems.

In this section, the term *membrane system* is used to refer any cell-like P systems introduced at the previous sections. An arbitrary membrane system of the order q, $q \geq 1$, will be described by a tuple

$$(\Gamma, \Gamma_0, \Gamma_1, \mathcal{E}, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, \mathcal{P}_\mathcal{R}, i_{out})$$

where we can think that $\Gamma_0 = \Gamma_1 = \emptyset$ for membrane systems without separation rules, $\mathcal{E} = \emptyset$ for basic transition P systems or P systems with active membranes, and $\mathcal{P}_\mathcal{R} \neq \emptyset$ only for basic transition P systems.

Next, we introduce the concept of recognizer associated with the membrane systems defined in the previous section.

Definition 7. *A recognizer membrane system*

$$\Pi = (\Gamma, \Gamma_0, \Gamma_1, \mathcal{E}, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$$

is a membrane system verifying the following:

- *The working alphabet Γ has two distinguished objects* `yes` *and* `no`*, with at least one copy of them present in some initial multisets, but none of them initially present in $\mathcal{E}$;*
- *there exists an additional alphabet Σ (the input alphabet) strictly contained in Γ such that $\mathcal{E} \subseteq \Gamma \setminus \Sigma$;*
- *$\mathcal{M}_1, \ldots, \mathcal{M}_q$ are multisets over $\Gamma \setminus \Sigma$;*
- *$i_{in} \in \{1, \ldots, q\}$ is the label of the input membrane;*
- *the output zone i_{out} is the environment;*
- *all computations halt;*
- *if $\mathcal{C}$ is a computation of Π, then either object* `yes` *or object* `no` *(but not both) must have been released into the environment, and only at the last step of the computation.*

For each multiset $\mathcal{M}$ over the input alphabet Σ, a *computation* of Π with input multiset $\mathcal{M}$ starts from the configuration of the form $(\mu, \mathcal{M}_1, \ldots, \mathcal{M}_{i_{in}} + \mathcal{M}, \ldots, \mathcal{M}_q, \emptyset)$, where the input multiset $\mathcal{M}$ has been added to the content of the input membrane i_{in}. That is, we have an initial configuration associated with each input multiset $\mathcal{M}$ over Σ in recognizer membrane systems. We denote by $\Pi + \mathcal{M}$ the P system Π with input multiset $\mathcal{M}$.

3.6 Polynomial complexity classes of recognizer membrane systems

Next, according to [27], we define what solving a decision problem by a family of recognizer P systems with symport/antiport rules, *in a uniform way*, means.

Definition 8. *A decision problem $X = (I_X, \theta_X)$ is solvable in polynomial time by a family $\mathbf{\Pi} = \{\Pi(n) \mid n \in \mathbb{N}\}$ of recognizer membrane systems (in a uniform way) if the following conditions hold:*

- *the family $\mathbf{\Pi}$ is polynomially uniform by Turing machines, that is, there exists a deterministic Turing machine working in polynomial time which constructs the system $\Pi(n)$ from $n \in \mathbb{N}$ (n expressed in unary);*
- *there exists a pair (cod, s) of polynomial-time computable functions over I_X such that:*
  - *for each instance $u \in I_X$, $s(u)$ is a natural number and $cod(u)$ is an input multiset of the system $\Pi(s(u))$;*
  - *for each $n \in \mathbb{N}$, $s^{-1}(n)$ is a finite set;*
  - *the family $\mathbf{\Pi}$ is polynomially bounded with regard to (X, cod, s), that is, there exists a polynomial function p, such that for each $u \in I_X$ every computation of $\Pi(s(u)) + cod(u)$ is halting and it performs at most $p(|u|)$ steps;*
  - *the family $\mathbf{\Pi}$ is sound with regard to (X, cod, s), that is, for each $u \in I_X$, if there exists an accepting computation of $\Pi(s(u)) + cod(u)$, then $\theta_X(u) = 1$;*
  - *the family $\mathbf{\Pi}$ is complete with regard to (X, cod, s), that is, for each $u \in I_X$, if $\theta_X(u) = 1$, then every computation of $\Pi(s(u)) + cod(u)$ is an accepting one.*

According to Definition 8, we say that for each $u \in I_X$, the recognizer membrane system $\Pi(s(u)) + cod(u)$ is *confluent*, in the sense that all possible computations of the system must give the same answer.

If $\mathcal{R}$ is a class of recognizer membrane systems, then we denote by $\mathbf{PMC}_{\mathcal{R}}$ the set of all decision problems which can be solved in polynomial time (and in a uniform way) by means of systems from $\mathcal{R}$. The class $\mathbf{PMC}_{\mathcal{R}}$ is closed under complement and polynomial-time reductions (see [27] for details).

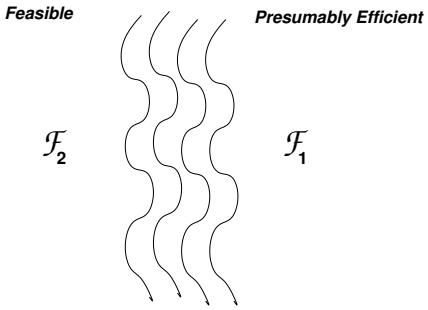
4 Frontiers of tractability in membrane systems

We say that a class of recognizer membrane systems $\mathcal{F}$ is *presumably efficient* if there exists an **NP**-complete problem that can be solved in polynomial time by a family of systems from $\mathcal{F}$. From the properties of the **NP**-completeness, we deduce that *any* **NP**-complete problem can be solved in polynomial time by families of a presumably efficient class of recognizer membrane systems. Because class $\mathbf{PMC}_{\mathcal{F}}$ is closed under complement and polynomial-time reductions (see [27] for details), if the class $\mathcal{F}$ is presumably efficient then $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{F}}$.

We say that a class of recognizer membrane systems $\mathcal{F}$ is *feasible* if only tractable problems can be solved in polynomial time by a family of systems from $\mathcal{F}$, that is, if $\mathbf{PMC}_{\mathcal{F}} = \mathbf{P}$. According to these definitions, if $\mathbf{P} = \mathbf{NP}$ then a class $\mathcal{F}$ is feasible if and only if it is presumably efficient. Besides, if $\mathbf{P} \neq \mathbf{NP}$ then each feasible class is not presumably efficient. Nevertheless, under that hypothesis a non-feasible class could be non-presumably efficient (as a consequence of the

Ladner theorem by which if $\mathbf{P} \neq \mathbf{NP}$ then there exist **NP**-intermediate problems, that is, problems which are neither in the class **P** nor in the class of **NP**-complete problems, see [11] for details).

Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two models from a computing paradigm such that $\mathcal{F}_1$ is an *extension* of $\mathcal{F}_2$, in other words, $\mathcal{F}_1$ is obtained from $\mathcal{F}_2$ by adding some syntactic or semantic ingredients (called *additional ingredients*). In this case, each solution of a decision problem in model $\mathcal{F}_2$ is also a solution in model $\mathcal{F}_1$. In this context, if $\mathcal{F}_2$ is a feasible model and $\mathcal{F}_1$ is a presumably efficient model, then we say that the additional ingredients provide a *frontier of tractability* between tractability and **NP**-hardness.



Let us consider two models $\mathcal{F}_1$ and $\mathcal{F}_2$ of recognizer membrane systems such that $\mathcal{F}_2$ is feasible, $\mathcal{F}_1$ is presumably efficient and model $\mathcal{F}_1$ is an extension of model $\mathcal{F}_2$. On the one hand, *translating* an efficient solution of an **NP**-complete problem by a family of systems in $\mathcal{F}_1$, into an efficient solution by a family of systems in $\mathcal{F}_2$ amounts to proving **P**=**NP**. On the other hand, proving that without the additional ingredients in $\mathcal{F}_1$ it is not possible to solve an **NP**-complete problem in polynomial-time, then the result $\mathbf{P} \neq \mathbf{NP}$ follows. Hence, each frontier of tractability provide a tool to tackle the **P** versus **NP** problem.

4.1 Basic transition P systems

Let us recall that the decision problem associated with a Turing machine M with input alphabet Σ_M is the problem $X_M = (I_M, \theta_M)$, where $I_M = \Sigma_M^*$, and for every $w \in \Sigma_M^*$, $\theta_M(w) = 1$ if and only if M accepts w. Then we say that a Turing machine M is simulated in polynomial time by a family of recognizer membrane systems from $\mathcal{R}$ if $X_M \in \mathbf{PMC}_{\mathcal{R}}$.

In [9] an efficient simulation of deterministic Turing machines by recognizer basic transition P systems was given.

Proposition 1. (`Sevilla theorem`) *Every deterministic Turing machine working in polynomial time can be simulated in polynomial time by a family of recognizer basic transition P systems.*

Also, in [9] was shown that each confluent basic transition P system can be (efficiently) simulated by a deterministic Turing machine. As a consequence, if a decision problem is solvable in polynomial time by a family of recognizer basic transition P systems, then there exists a deterministic Turing machine solving it in polynomial time. Then, we have the following result: $\mathbf{P} = \mathbf{PMC}_{\mathcal{T}}$, being $\mathcal{T}$ the class of all recognizer basic transition P systems. Thus, the use of cooperative rules in basic transition P systems and their ability to create exponential workspace (in terms of number of objects) in linear time is not enough to efficiently solve **NP**-complete problems (assuming that $\mathbf{P} \neq \mathbf{NP}$).

4.2 P systems with active membranes

Let us denote by $\mathcal{DAM}$ the class of all recognizer P systems with active membranes and let $\mathcal{NAM}$ be the class of recognizer P systems with active membranes which do not make use of division rules.

In the framework of cell-like membrane systems, confluent systems from $\mathcal{NAM}$ can be efficiently simulated by a deterministic Turing machine [43].

Proposition 2. (`Milano theorem`) *A deterministic P system with active membranes but without membrane division can be simulated by a deterministic Turing machine with a polynomial slowdown.*

As a consequence of the Milano theorem, we have $\mathbf{PMC}_{\mathcal{NAM}} \subseteq \mathbf{P}$. Bearing in mind that the reverse implication is easily deduced from Definition 8, we have $\mathbf{PMC}_{\mathcal{NAM}} = \mathbf{P}$.

By using membrane systems from $\mathcal{DAM}$ which do not make use of dissolution rules, different efficient solutions to strongly **NP**-complete problems (`SAT` [27], `Clique` [3], `Bin Packing` [29], `Common Algorithmic Problem` [30]) have been given. Since the class $\mathbf{PMC}_{\mathcal{DAM}}$ is closed under complement and polynomial-time reductions, we deduce that $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{DAM}}$.

Remark 1: In the framework of P systems with active membranes, the use or not of the division rules provides a borderline for the tractability of decision problems, assuming that $\mathbf{P} \neq \mathbf{NP}$. Thus, by using division rules we can solve **NP**-complete problems in polynomial time, but without division rules only problems in **P** can be solved in an efficient way. Then, in this framework, **cooperative rules are not necessary** to obtain frontiers of tractability.

4.3 Polarizationless P systems with active membranes

Let us denote by $\mathcal{DAM}^0(\alpha, \beta, \gamma, \eta)$ the class of all recognizer polarizationless P systems with active membranes which make use of division rules such that:

- If $\alpha = +e$ ($\alpha = -e$, resp.) then object-evolution rules are permitted (forbidden, resp.);

- if $\beta = +c$ ($\beta = -c$, resp.) then communication rules are permitted (forbidden, resp.);
- if $\gamma = +d$ ($\alpha = -d$, resp.) then dissolution rules are permitted (forbidden, resp.); and
- if $\eta = +n$ ($\eta = -n$, resp.) then division rules for elementary and non-elementary membranes are permitted (only division rules for non-elementary membranes are permitted).

In the same way, we denote by $\mathcal{SAM}^0$ the corresponding class when separation rules are considered instead of division rules.

The so-called Păun's conjecture can be formally formulated in terms of membrane computing complexity classes as follows: $\mathbf{P} = \mathbf{PMC}_{\mathcal{DAM}^0(+e,+c,+d,-n)}$. Currently, this is a relevant open problem. However, several partial solution have been given.

Let Π be a recognizer polarizationless P system with active membranes which does not make use of dissolution rules. A directed graph (called *dependency graph*) can be associated with Π verifying the following property: *every accepting computation of Π is characterized by the existence of a path in the graph between two specific nodes*. Based in this concept and by using the fact that *reachability problem* is in class **P**, the following result has been provided [8]:

$$\mathbf{P} = \mathbf{PMC}_{\mathcal{DAM}^0\,(+e,+c,-d,-n)} = \mathbf{PMC}_{\mathcal{DAM}^0\,(+e,+c,-d,+n)}$$

Thus, polarizationless P systems with active membranes which do not make use of dissolution rules cannot solve **NP**-complete problems in polynomial time (unless **P**=**NP**). This result can be considered as a partial affirmative answer to the Păun's conjecture.

Let us now consider polarizationless P systems with active membranes making use of dissolution rules. Will it be possible to solve **NP**-complete problems in that framework? Several authors [1, 8] gave a positive answer when division for non-elementary membranes are allowed. The mentioned papers provide solutions in linear time to `SAT` problem and `Subset Sum` problem, respectively. Hence, we have $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{DAM}^0\,(+e,+c,+d,+n)}$. Therefore, a *partial negative* answer to Păun's conjecture is given: assuming that $\mathbf{P} \neq \mathbf{NP}$ and making use of dissolution rules and division rules for elementary and non-elementary membranes, computationally hard problems can be efficiently solved avoiding polarizations. The answer is partial because efficient solvability of **NP**-complete problems by means of families from $\mathcal{DAM}^0\,(+e,+c,+d,n)$ is unknown.

Remark 2: In the framework of polarizationless P systems with active membranes, the use or not of dissolution rules provides a borderline for the tractability of decision problems, assuming that $\mathbf{P} \neq \mathbf{NP}$, that is, by using dissolution rules we can solve **NP**-complete problems in polynomial time, but without dissolution rules only problems in **P** can be solved in an efficient way.

4.4 Cooperation in polarizationless P systems with active membranes

The role of dissolution rules in the framework of $\mathcal{DAM}^0$ is crucial in order to provide polynomial-time solutions to computationally hard problems (assuming that $\mathbf{P} \neq \mathbf{NP}$). In this section we prove that by using (very restrictive) cooperative rules in polarizationless P systems with active membranes, it is possible to solve **NP**-complete problems in an efficient way.

Next, several types of *minimal cooperation* in object-evolution rules are considered in the framework of polarizationless P systems with active membranes. The term "minimal cooperation" is used in the following sense: the left-hand side of such rules consists of two symbols.

- *Minimal cooperation* (**mc**): object-evolution rules are of the form $[\, u \rightarrow v \,]_h$, where $u, v \in M(\Gamma)$ such that $1 \leq |u| \leq 2$.
- *Primary minimal cooperation* (**pmc**): object- evolution rules are of the form $[\, u \rightarrow v \,]_h$, where $u, v \in M(\Gamma)$ and $1 \leq |u|, |v| \leq 2$.
- *Bounded minimal cooperation* (**bmc**): object- evolution rules are of the form $[\, u \rightarrow v \,]_h$, where $u, v \in M(\Gamma)$ and $1 \leq |v| \leq |u| \leq 2$.
- *Minimal cooperation and minimal production* (**mcmp**): object- evolution rules are of the forms $[\, a \rightarrow b \,]_h$ or $[\, a\, b \rightarrow c \,]_h$, where $a, b, c \in \Gamma$.

In polarizationless P systems with active membranes and minimal cooperation in object-evolution rules, the remaining rules (send-in communication rules, send-out communication, dissolution and division) are non-cooperative rules. Besides, the rules are applied according to the same principles than in "classical" P systems with active membranes.

In the expression $\mathcal{DAM}^0(\alpha, \beta, \gamma, \delta)$, parameter α associated with object-evolution rules is extended as follows:

- if $\alpha = mc$ then minimal cooperation in object- evolution rules are permitted.
- if $\alpha = pmc$ then primary minimal cooperation in object- evolution rules are permitted.
- if $\alpha = bmc$ then bounded minimal cooperation in object- evolution rules are permitted.
- if $\alpha = mcmp$ then minimal cooperation and minimal production in object-evolution rules are permitted.

Next, we summarize some interesting results.

1. Families of systems from $\mathcal{DAM}^0(+e, +c, +d, +n)$ can solve **PSPACE**-complete problems in polynomial time, that is, $\mathbf{PSPACE} \subseteq \mathbf{PMC}_{\mathcal{DAM}^0(+e,+c,+d,+n)}$ [5]. In fact, $\mathbf{PSPACE} = \mathbf{PMC}_{\mathcal{DAM}^0(+e,+c,+d,+n)}$ (see [36] and [37] for details).
2. Families of systems from $\mathcal{DAM}^0(+e, +c, -d, +n)$ can efficiently solve only problems in class **P**, that is, $\mathbf{PMC}_{\mathcal{DAM}^0(+e,+c,-d,+n)} = \mathbf{P}$ (see [8] for details).

3. Families of systems from $\mathcal{SAM}^0(+e,+c,-d,+n)$ can efficiently solve only problems in class **P**, that is, $\mathbf{PMC}_{\mathcal{SAM}^0(+e,+c,-d,+n)} = \mathbf{P}$ (see [42] for details).
4. Families of systems from $\mathcal{DAM}^0(bmc,+c,-d,-n)$ can solve **NP**-complete problems in polynomial time, i.e., $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{DAM}^0(bmc,+c,-d,-n)}$ (see [38] for details).
5. Families of systems from $\mathcal{SAM}^0(bmc,+c,+d,+n)$ can efficiently solve only problems in class **P**, that is, $\mathbf{PMC}_{\mathcal{SAM}^0(bmc,+c,+d,+n)} = \mathbf{P}$ (see [40] for details).
6. Families of systems from $\mathcal{SAM}^0(pmc,+c,-d,-n)$ can solve **NP**-complete problems in polynomial time, i.e., $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{SAM}^0(pmc,+c,-d,-n)}$ (see [38] for details).
7. Families of systems from $\mathcal{DAM}^0(mcmp,+c,-d,-n)$ can solve **NP**-complete problems in polynomial time, i.e., $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{DAM}^0(mcmp,+c,-d,-n)}$ (see [41] for details).
8. Families of systems from $\mathcal{SAM}^0(mcmp,+c,+d,+n)$ can efficiently solve only problems in class **P**, that is, $\mathbf{P} = \mathbf{PMC}_{\mathcal{SAM}^0(mcmp+c,+d,+n)}$ (see [41] for details).

From these results, the following frontiers of tractability are obtained.

- In the framework $\mathcal{DAM}^0(*,+c,-d,-n)$: passing from non-cooperative object-evolution rules to bounded minimal cooperation in object-evolution rules.
- In the framework $\mathcal{SAM}^0(*,+c,-d,-n)$: passing from non-cooperative object-evolution rules to primary minimal cooperation in object-evolution rules.
- In the framework $*\mathcal{AM}^0(bmc,+c,-d,-n)$: passing from separation rules to division rules.
- In the framework $*\mathcal{AM}^0(mcmp,+c,-d,-n)$: passing from separation rules to division rules.

Non Efficiency (*Feasible*)	Efficiency (*Presumably Efficient*)	Frontiers
$\mathcal{DAM}^0(+e,+c,-d,-n)$	$\mathcal{DAM}^0(mcmp,+c,-d,-n)$	*minimal cooperation and minimal production*
$\mathcal{SAM}^0(+e,+c,-d,-n)$	$\mathcal{SAM}^0(pmc,+c,-d,-n)$	*primary minimal cooperation*
$\mathcal{SAM}^0(bmc,+c,-d,-n)$	$\mathcal{DAM}^0(bmc,+c,-d,-n)$	*separation vs division*
$\mathcal{SAM}^0(mcmp,+c,-d,-n)$	$\mathcal{DAM}^0(mcmp,+c,-d,-n)$	*separation vs division*

4.5 P systems with symport/antiport rules

The class of all recognizer P systems with symport/antiport rules and with membrane division (resp. membrane separation) will be denoted by **CDC** (resp. **CSC**). For each natural number $k \geq 1$, we denote by $\mathbf{CDC}(k)$ (resp. $\mathbf{CSC}(k)$) the class of all recognizer P systems with membrane division (resp. membrane separation) and with symport/antiport rules of length at most k. In the case of P systems without environment, we denote by $\widehat{\mathbf{CDC}}$, $\widehat{\mathbf{CSC}}$, $\widehat{\mathbf{CDC}}(k)$ and $\widehat{\mathbf{CSC}}(k)$, respectively, the corresponding class. Obviously, recognizer P systems from $\mathbf{CDC}(1)$, $\mathbf{CSC}(1)$, $\widehat{\mathbf{CDC}}(1)$ and $\widehat{\mathbf{CSC}}(1)$, are non-cooperative systems, and the remaining recognizer P systems are cooperative systems.

Next, we summarize some interesting results.

- Families of non-cooperative P systems with symport/antiport rules can only efficiently solve problems in class **P**, that is, $\mathbf{P} = \mathbf{PMC}_{CDC(1)} = \mathbf{PMC}_{CSC(1)}$ (see [14] for details).
- In [39] a family of P systems with division rules and symport/antiport rules using minimal cooperation solving in polynomial-time the `HAM-CYCLE` problem, a well known **NP**-complete problem [7], has been given. Thus, $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{CDC(2)}$.
- Families of P systems with separation rules and symport/antiport rules using minimal cooperation can only efficiently solve problems in class **P**, that is, $\mathbf{P} = \mathbf{PMC}_{CSC(2)}$ (see [13] for details). However, in the cited paper, a family of P systems with separation rules and using symport/antiport rules with length at most three solving in polynomial-time the `SAT` problem, has been given, that is, $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{CSC(3)}$.
- The role of the environment is irrelevant when we try to provide polynomial-time solutions to **NP**-complete problems by means of families of P systems with symport/antiport rules and membrane division. Specifically, for each $k \in \mathcal{N}$ we have $\mathbf{PMC}_{\mathbf{CDC}(k+1)} = \mathbf{PMC}_{\widehat{\mathbf{CDC}}(k+1)}$ (see [15] for details).
- Families of P systems with symport/antiport rules and separation rules but without environment, can only efficiently solve problems in class **P**, that is, $\mathbf{P} = \mathbf{PMC}_{\widehat{\mathbf{CSC}}}$ (see [12] for details). Hence, the role of the environment is relevant when we try to provide polynomial-time solutions to **NP**-complete problems by means of families of P systems with symport/antiport rules and membrane separation

From these results, the following frontiers of tractability are obtained.

Non Efficiency (*Feasible*)	Efficiency (*Presumably Efficient*)	Frontiers
$\mathcal{CDC}(1)$	$\mathcal{CDC}(2)$	(*length*)
$\mathcal{CSC}(2)$	$\mathcal{CSC}(3)$	(*length*)
$\mathcal{CSC}(2)$	$\mathcal{CDC}(2)$	(*separation vs division*)
$\widehat{\mathcal{CSC}(2)}$	$\widehat{\mathcal{CDC}(2)}$	(*separation vs division*)
$\widehat{\mathcal{CSC}}$	$\mathcal{CSC}$	(*environment*)

5 Satisfiability problems and cooperation in rewriting rules

In this section, a relationship among cooperative rewriting rules and instances of 2–SAT problem and 3–SAT problem is highlighted.

5.1 Implication graph associated with instances of 2–SAT

Let $\varphi = C_1 \wedge \cdots \wedge C_p$ be a boolean formula consisting of p ($p \geq 2$) clauses $C_j = l_j^1 \vee l_j^2,\ 1 \leq j \leq p$, that is, φ is an instance of 2–SAT. The *implication graph* $G_\varphi = (V_\varphi, E_\varphi)$ is the directed graph defined as follows:

- V_φ is the set of all literals associated with the set $Var(\varphi)$ of variables of φ.
- $E_\varphi \subseteq V_\varphi \times V_\varphi$ is the following set of arcs: $(x, y) \in V_\varphi \times V_\varphi$ if and only if there exists a clause $C = l^1 \vee l^2$ of φ such that $x = \overline{l^1} \wedge y = l^2$ or $x = \overline{l^2} \wedge y = l^1$.

According with the previous definition, each clause $C_j = l_j^1 \vee l_j^2$ of φ has associated two arcs $(\overline{l_j^1}, l_j^2)$ and $(\overline{l_j^2}, l_j^1)$. In some sense, these arcs capture the boolean values of logical implications associated with each clause of φ in a natural way.

It is worth pointing out that the implication graphs associated with instances of 2–SAT verify the following result (see [20] for details):

Theorem 1. *Let $\varphi = C_1 \wedge \cdots \wedge C_p$ be an instance of* 2–SAT *with p ($p \geq 2$) clauses. Then the following assertions are equivalent:*

- *φ is unsatisfiable.*
- *There exists a variable $x \in Var_\varphi$ such that there are paths in G_φ from x to $\overline{x}$ and from $\overline{x}$ to x.*

From this theorem, by using the tractability of the reachability problem, it is easy to follow that 2–SAT is a problem in class **P**.

5.2 Rewriting rules associated with instances of k–SAT

Let $k \geq 2$ and $\varphi = C_1 \wedge \cdots \wedge C_p$ be a boolean formula consisting of p ($p \geq 2$) clauses $C_j = l_j^1 \vee \ldots \vee l_j^k,\ 1 \leq j \leq p$, that is, φ is an instance of k–SAT. Let us recall that each clause C_j is logically equivalent to the boolean formula $\overline{l_j^1} \wedge \cdots \wedge \overline{l_j^{k-1}} \rightarrow l_j^k$. If σ is a truth assignment which makes true the formula φ and $\sigma(\overline{l_j^1} \wedge \cdots \wedge \overline{l_j^{k-1}}) = 1$ then we deduce that $\sigma(l_j^k) = 1$.

In this context, the expression $\overline{l_j^1} \cdots \overline{l_j^{k-1}} \rightarrow l_j^k$ is associated with clause C_j and it can be viewed as a rewriting rule in the following sense: the rule is applicable for a truth assignment σ which makes true the formula φ if and only if $\sigma(\overline{l_j^1} \wedge \cdots \wedge \overline{l_j^{k-1}}) = 1$. The application of that rule produces $\sigma(l_j^k) = 1$ as some kind of "information".

Following the same idea and bearing in mind the properties of logical equivalence of boolean formulas, different rewriting rules could be associated with formula φ. Specifically, for each t, $1 \leq t \leq k$, rewriting rules of the type $\overline{l_j^1} \cdots \widehat{\overline{l_j^t}} \cdots \overline{l_j^k} \rightarrow l_j^t$ are associated to clause C_j, where the expression $\overline{l_j^1} \cdots \widehat{\overline{l_j^t}} \cdots \overline{l_j^k}$ means that literal $\overline{l_j^t}$ does not appear in the left-hand side. In this paper, the particular case $t = k$ has been considered.

Next, we show that any instance of 2–SAT problem has associated non-cooperative rewriting rules (their left-hand side contain only one object) and any instance of 3–SAT problem has associated cooperative rewriting rules (their left-hand side contain at least two objects).

Instances of 2–SAT

Let $\varphi = C_1 \wedge \cdots \wedge C_p$ be a boolean formula consisting of p ($p \geq 2$) clauses $C_j = l_j^1 \vee l_j^2,\ 1 \leq j \leq p$, that is, φ is an instance of 2–SAT. Then, clause $l_j^1 \vee l_j^2$ has associated the "non-cooperative rewriting rule" $\overline{l_j^1} \rightarrow l_j^2$. As usual, the term "non-cooperative" refers to the propery that its left-hand side contains only one "object" (in this case, one literal).

We say that the non-cooperative rewriting rule $\overline{l_j^1} \rightarrow l_j^2$ is *applicable for a truth assignment* σ associated with the set of variables Var_φ of φ if $\sigma(\overline{l_j^1}) = 1$; that is, in order to determine if such a rule is applicable for σ only is necessary to know the truth value of one literal. When applying a rule $\overline{l_j^1} \rightarrow l_j^2$ for a truth assignment σ, we deduce the following information: $\sigma(l_j^2) = 1$.

Instances of 3–SAT

Let $\varphi = C_1 \wedge \cdots \wedge C_p$ be a boolean formula consisting of p ($p \geq 2$) clauses $C_j = l_j^1 \vee l_j^2 \vee l_j^3,\ 1 \leq j \leq p$, that is, φ is an instance of 3–SAT. We associate with clause $l_j^1 \vee l_j^2 \vee l_j^3$ the "cooperative rewriting rule" $\overline{l_j^1}\ \overline{l_j^2} \rightarrow l_j^3$. As usual, the term

"cooperative" refers to the propery that its left-hand side contains more than one "object" (more than one literal).

We say that cooperative rewriting rule $\overline{l_j^1}\ \overline{l_j^2} \rightarrow l_j^3$ is *applicable for a truth assignment* σ associated with the set of variables Var_φ of φ if $\sigma(\overline{l_j^1}) = 1$ and $\sigma(\overline{l_j^2}) = 1$; that is, in order to determine if such a rule is applicable for σ is necessary to know the truth value of two literals. So, these literals must "cooperate" in order to apply the rewriting rule. When applying a rule $\overline{l_j^1}\ \overline{l_j^2} \rightarrow l_j^3$ for a truth assignment σ, we deduce the following information: $\sigma(l_j^3) = 1$.

Cooperation as a new frontier of tractability

It is well known that 2–SAT $\in$ **P** and 3–SAT is an **NP**-complete problem. On the one hand, non-cooperative rewriting rules have been associated with instances of 2–SAT and cooperative rewriting rules have been associated with instances of 3–SAT. On the other hand, "passing" from 2–SAT to 3–SAT can be interpreted as passing from tractability to (the presumable) intractability (assuming that **P** $\neq$ **NP**). In this context we can consider that passing from non-cooperative rewriting rules to cooperative ones amounts to passing from tractability to (the presumable) intractability (assuming that **P** $\neq$ **NP**).

6 Conclusions

The quest for tools that provide new approaches to address the problem P versus NP, is a major challenge in computer science due to the relevance of the above mentioned problem. This paper focuses on tools related to frontiers of tractability expressed in terms of syntactic or semantic ingredients associated with models in a computing paradigm.

The role of cooperation of objects to trigger rewriting rules is analysed in order to obtain this kind of borderlines in the framework of Membrane Computing. Specifically, some cell-like membrane systems have been considered:

- Basic transition P systems.
- P systems with active membranes -with/without electrical charges and with/ without environment- and with membrane division or membrane separation.
- P systems with symport/antiport rules -with or without environment- and with membrane division or membrane separation.

References

1. A. Alhazov, L. Pan and Gh. Păun. Trading polarizations for labels in P systems with active membranes. *Acta Informaticae*, **41**, 2-3(2004), 111-144.

2. A. Alhazov, L. Pan. Polarizationless P systems with active membranes. *Grammars*, **7** (2004), 141-159.
3. A. Alhazov, L. Pan, Gh. Păun. Trading polarizations for labels in P systems with active membranes. *Acta Informaticae*, **41**, 2-3 (2004), 111-144.
4. A. Alhazov, R. Freund, Gh. Păun. P systems with active membranes and two polarizations. In Gh. Păun, A. Riscos-Núñez, Á. Romero, F. Sancho (eds.) *Proceedings of the Second Brainstorming Week on Membrane Computing*, Sevilla, 2-7 February 2004, Research Group on Natural Computing, TR 01/2004, Universidad de Sevilla, Fénix Editora, 2004, pp. 2035.
5. A. Alhazov, M.J. Pérez-Jiménez. Uniform solution of QSAT using polarizationless active membranes. *Lecture Notes in Computer Science*, **4664** (2007), 122-133.
6. T.H. Cormen, C.E. Leiserson, R.L. Rivest. *An Introduction to Algorithms*. The MIT Press, Cambridge, Massachussets, 1994.
7. M.R. Garey, D.S. Johnson. *Computers and Intractability A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.
8. M.Á. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Núñez, F.J. Romero-Campero. On the power of dissolution in P systems with active membranes. *Lecture Notes in Computer Science*, **3850** (2006), 224-240.
9. M.Á. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Núñez, F.J.Romero-Campero and Á. Romero Characterizing tractability by cell-like membrane systems. In K.G. Subramanian, K. Rangarajan, M. Mukund (eds.) *Formal models, languages and applications*, World Scientific, Singapore, 2006, pp. 137-154.
10. M. Ionescu, Gh. Păun, T. Yokomori. Spiking neural P systems, *Fundamenta Informaticae*, **71**, 2-3 (2006), 279-308.
11. R.E. Ladner. On the structure of polynomial time reducibility. *Journal of the ACM*, **22**, 1 (1975), 155171.
12. LF. Macias-Ramos, M.J. Pérez-Jiménez, A. Riscos-Núñez, L. Valencia-Cabrera. Membrane fission versus cell division: when membrane proliferation is not enough. *Theoretical Computer Science*, **608** (2015), 57-65.
13. L.F. Macias-Ramos, B. Song, L. Valencia-Cabrera, L. Pan, M.J. Pérez-Jiménez. Membrane fission: A computational complexity perspective. *Complexity*, **21**, 6 (2016), 321-334
14. L.F. Macias-Ramos, B. Song, T. Song, L. Pan, M.J. Pérez-Jiménez. Limits on efficient computation in P systems with symport/antiport. In C. Graciani, Gh. Păun, A. Riscos-Núñez, L. Valencia-Cabrera (eds.) *Proceedings of the Fifteenth Brainstorming Week on Membrane Computing*, January 31, February 3, 2017, Research Group on Natural Computing, TR 01/2017, Universidad de Sevilla, Fénix Editora, 2017, pp. 147-160
15. L.F. Macias-Ramos, B. Song, T. Song, L. Pan, M.J. Pérez-Jiménez. On efficiency of P systems with symport/antiport and membrane division. In C. Graciani, Gh. Păun, A. Riscos-Núñez, L. Valencia-Cabrera (eds.) *Proceedings of the Fifteenth Brainstorming Week on Membrane Computing*, January 31, February 3, 2017, Research Group on Natural Computing, TR 01/2017, Universidad de Sevilla, Fénix Editora, 2017 pp. 129-146.
16. C. Martín-Vide, J. Pazos, Gh. Păun, A. Rodriguez-Patón. Tissue P systems, *Theoretical Computer Science*, **296**, 2 (2003), 295-326.
17. L. Pan, T.-O. Ishdorj. P systems with active membranes and separation rules. *Journal of Universal Computer Science*, **10**, 5 (2004), 630-649.

18. L. Pan, A. Alhazov, Solving HPP and SAT by P systems with active membranes and separation rules. *Acta Informaticae*, **43** (2006), 131-145.
19. L. Pan, A. Alhazov, T.-O. Ishdorj, Further remarks on P systems with active membranes, separation, merging, and release rules. *Soft Computing*, **8**, (2004), 1-5.
20. Ch. H. Papadimitriou. *Computational Complexity*, Addison-Wesley Publishing Company, Reading, Massachussets, USA, 1994.
21. Gh. Păun, Computing with membranes, *Journal of Computer System Sciences*, **61**, 1 (2000), 108-143.
22. Gh. Păun. Attacking **NP**-complete problems. In *Unconventional Models of Computation, UMC'2K* (I. Antoniou, C. Calude, M. J. Dinneen, eds.), Springer-Verlag, 2000, pp. 94-115.
23. A. Păun, Gh. Păun. The power of communication: P systems with symport/antiport, *New Generation Computing*, **20**, 3 (2002), 295-305.
24. Gh. Păun: *Membrane Computing. An introduction.* Springer-Verlag, Berlin, 2002.
25. Gh. Păun, M.J. Pérez-Jiménez, A. Riscos-Núñez. Tissue P System with cell division. *International Journal of Computers, Communications and Control*, **3**, 3 (2008), 295-303.
26. Gh. Păun, G. Rozenberg, A. Salomaa (eds.). *The Oxford Handbook of Membrane Computing*, Oxford University Press, Oxford, 2010.
27. M.J. Pérez-Jiménez, Á. Romero, F. Sancho. Complexity classes in models of cellular computing with membranes. *Natural Computing*, **2**, 3 (2003), 265-285.
28. M.J. Pérez-Jiménez, F.J. Romero-Campero: Trading polarizations for bi-stable catalysts in P systems with active membranes. *Lecture Notes in Computer Science*, **3365** (2005), 373-388.
29. M.J. Pérez-Jiménez and F.J. Romero-Campero. An efficient family of P systems for packing items into bins. *Journal of Universal Computer Science*, **10**, 5 (2004), 650-670.
30. M.J. Pérez-Jiménez and F.J. Romero-Campero. Attacking the Common Algorithmic Problem by recognizer P systems. *Lecture Notes in Computer Science*, **3354** (2005), 304-315.
31. M.J. Pérez-Jiménez, Á. Romero, F. Sancho. A polynomial complexity class in P systems using membrane division. *Journal of Automata, Languages and Combinatorics*, **11**, 4, (2006), 423-434.
32. M.J. Pérez-Jiménez, Á. Romero, F. Sancho, A polynomial complexity class in P systems using membrane division, *Journal of Automata, Languages and Combinatorics*, **11**, 4 (2006) 423-434.
33. M.J. Pérez-Jiménez. The P versus NP problem from Membrane Computing view. *European Review*, **22**, 1 (2014), 18-33.
34. M.J. Pérez-Jiménez, P. Sosik. An Optimal Frontier of the Efficiency of Tissue P Systems with Cell Separation. *Fundamenta Informaticae*, **138**, 1-2 (2015), 45-60
35. A.E. Porreca, G. Mauri, C. Zandron. Complexity classes for membrane systems. *Informatique théorique et applications*, **40**, 2 (2006), 141-162.
36. P. Sosik, A. Rodriguez-Patón. Membrane Computing and Complexity Theory: A Characterization of PSPACE. *J. Comput. System Sci.*, **73**, 1 (2007), 137-152.
37. P. Sosik. PSPACE Limits the Power of P Systems with Active Membranes. *Journal of Automata, Languages and Combinatorics*, **19**, 1-4 (2014), 291-304.
38. L. Valencia-Cabrera, D. Orellana-Martín, M.Á. Martínez-del-Amor, A. Riscos-Núñez, M.J. Pérez-Jiménez. Polarizationless P systems with active membranes: Computational complexity aspects. *Journal of Automata, Languages and Combinatorics*, **21**, 1-2 (2016), 101-117.

39. L. Valencia-Cabrera, B. Song, L.F. Macias-Ramos, L. Pan, A. Riscos- Núñez, M.J. Pérez-Jiménez. Minimal cooperation in P systems with symport/antiport: A complexity approach. In L.F. Macias, Gh. Păun, A. Riscos-Núñez, L. Valencia (eds) *Proceedings of the Thirteenth Brainstorming Week on Membrane Computing*, 2-6 February, 2015, Research Group on Natural Computing, TR 01/2015, Universidad de Sevilla, Fénix Editora, 2015, pp. 301-323.
40. L. Valencia-Cabrera, D. Orellana-Martín, A. Riscos-Núñez, M.J. Pérez-Jiménez. Minimal cooperation in polarizationless P systems with active membranes. In C. Graciani et al. (eds.) *Proceedings of the Fourteenth Brainstorming Week on Membrane Computing*, Sevilla, 1-5 February 2016, Research Group on Natural Computing, TR 01/2016, Universidad de Sevilla, Fénix Editora, 2016, pp. 327-356.
41. L. Valencia-Cabrera, D. Orellana-Martín, M.Á. Martinez-del-Amor, A. Riscos-Núñez, M.J. Pérez-Jiménez. Reaching efficiency through collaboration in membrane systems: Dissolution, polarization and cooperation. *Theoretical Computer Science*, in press 2017 (doi: 10.1016/j.tcs.2017.04.015).
42. L. Valencia-Cabrera, D. Orellana-Martín, M.Á. Martínez-del-Amor, A. Riscos-Núñez, M.J. Pérez-Jiménez. Computational efficiency of minimal cooperation and distribution in polarizationless P systems with active membranes. *Fundamenta Informaticae*, **153**, 1-2 (2017), 147-172.
43. C. Zandron, C. Ferretti, G. Mauri. Solving NP-Complete Problems Using P Systems with Active Membranes. In I. Antoniou, C.S. Calude and M.J. Dinneen (eds.). *Unconventional Models of Computation, UMC'2K*, Springer-Verlag, 2000, pp. 289-301.
44. P systems website. `http://ppage.psystems.eu`.