The role of integral membrane proteins in computational complexity theory

David Orellana-Martín[1] · Miguel Á. Martínez-del-Amor[1] · Luis Valencia-Cabrera[1] · Agustín Riscos-Núñez[1] · Mario J. Pérez-Jiménez[1]

Abstract In the framework of Membrane Computing, several tools to tackle the **P** versus **NP** problems by means of frontiers of the efficiency expressed in terms of syntactic or semantic ingredients, have been developed. In this paper, an overview of the results in computational complexity theory concerning to membrane systems (tissue-like and cell-like approach) with symport/antiport rules (where objects are transported without evolving), is given. The frontiers are formulated regarding the length of communication rules, the kind of rules implementing the production of an exponential number of cells/membranes in polynomial time, and the role of the environment. An interesting remark of the obtained results refers that the underlying structure to membrane systems (directed graph versus rooted tree) does not matter in this context.

Keywords P systems · Computational complexity theory · Structure · Cell division · Membrane fission

1 Introduction

Membrane Computing, as a branch of Natural Computing, takes inspiration from the nature when developing mechanisms to solve problems. In this field, different kinds of bio-inspired devices are investigated, the so-called *membrane systems* or *P systems*, which are inspired by the structure and behavior of cells of living beings. Different variants of these systems have been studied, both inspired by the chemical reactions ocurring within cells in cell-like systems [1] or the transport of substances between cells in tissue-like or neural-like systems [2, 3]. For a comprehensive and wider explanation of these systems and their several variants, we invite the reader to see [4, 5]. Several kinds of fields, as economics, molecular biology, ecosystems and fault diagnosis, among others, have been solved in this framework [6, 7, 8]. Apart from these real-life applications, more theoretical fields can be addressed. From the universality of the systems to the classes of problems that can be *efficiently* solved by families of them, Membrane Computing represents a different perspective to deal with existing theories.

Membranes of living cells are composed primarily of lipids and proteins with a variable amount of carbohydrates attached to the surface. It makes the cytoplasmatic membrane a semipermeable or selectively permeable layer, letting certain molecules to pass through the membrane. Thanks to this, a membrane acts as a barrier for the cell, "defending" it from the environment. In fact, it is known that a high amount of genes code specifically for them [9].

The most common type of integral membrane proteins is the transmembrane protein, that is, proteins that pass through the membrane and "connect" the exterior of the cell with the insides of it. One of the main roles of this kind of protein is the passing of specific substances across the

✉ Mario J. Pérez-Jiménez
marper@us.es

David Orellana-Martín
dorellana@us.es

Miguel Á. Martínez-del-Amor
mdelamor@us.es

Luis Valencia-Cabrera
lvalencia@us.es

Agustín Riscos-Núñez
ariscosn@us.es

[1] Research Group on Natural Computing, Department of Computer Science and Artificial Intelligence, Universidad de Sevilla, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain

membrane. Depending on the protein, the substance and where is it coming from, to former one lets the last one pass or not.

Integral membrane proteins that let pass molecules across the membrane only in one direction are called symporters, and antiporters cotransport molecules or ions in opposite directions. This kind of behavior is crucial in some life processes [10].

In this sense, we can think about membrane systems replicating this behavior through symport/antiport rules. They were first introduced in tissue P systems, giving an abstraction of the transport of substance from a cell to another one or the interchange of molecules with their own environment, and later in cell-like systems [11]. These systems are inspired by the transport of substances within a cell between different organelles. Division rules [12, 13] and separation rules [14, 15] have been considered in these systems as a method to obtain an exponential amount of space in terms of regions in linear time and obtain efficient solutions to presumably intractable problems.

This paper is devoted to study the frontiers of efficiency obtained in terms of systems explained before, and tries to clarify the idea that when new frontiers between the complexity classes **P** and **NP** it is irrelevant if we are working with a P system viewed as a rooted tree or as a directed graph. The paper is structured as follows. Sects. 2 and 3 are devoted to introduce these models both syntactic and semantically, giving the classical definition and by adding division or separation rules. In Sect. 4 *recognizer* membrane systems are introduced as a computational device capable of solve decision problems. Next, some techniques used to prove the efficiency of recognizer membrane systems are explained.

2 Tissue-like P systems with symport/antiport rules

Inspired by the functioning of living cells in human tissues, tissue P systems are based on the transport of substances between cells, that is, the components initially placed in the system are not changed through the whole computation. It is important to remark that objects do not evolve in these systems, so new mechanisms must be implemented in order to solve computationally hard problems.

2.1 Syntax

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

1. Γ is a finite alphabet;
2. $\mathcal{E} \subsetneq \Gamma$;
3. $\mathcal{M}_1, \ldots, \mathcal{M}_q$ are multisets over Γ;
4. $\mathcal{R}$ is a finite set of communication rules of the form $(i,\ u\ /\ v,\ j)$, where u, v are multisets over Γ, some of them non-empty, and $i,j \in \{0, 1, \ldots, q\}$ with $i \neq j$.
5. $i_{out} \in \{0, 1, 2, \ldots, q\}$.

A tissue P system with symport/antiport rules$\Pi = (\Gamma, \mathcal{E}, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$ of degree $q \geq 1$ can be viewed as a set of q cells labelled by $1, \ldots, q$ such that: (a) $\mathcal{M}_1, \ldots, \mathcal{M}_q$ are multisets over the *working alphabet* representing the objects initially placed in the q cells 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) $\mathcal{R}$ is a finite set of communication rules over Γ; and (d) i_{out} 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 cell i in the case $1 \leq i \leq q$ and to refer to the environment in the case $i = 0$. A rule $(i,\ u\ /\ v,\ j)$ is called a *symport rule* if $u = \lambda$ or $v = \lambda$. A rule $(i,\ u\ /\ v,\ j)$ is called an *antiport rule* if $u \neq \lambda$ and $v \neq \lambda$. The length of rule $(i, u/\lambda, j)$ (resp., $(i,\ u\ /\ v,\ j)$) is defined as $|\ u\ |$ (resp., $|\ u\ | + |\ v\ |$). If $\mathcal{E} = \emptyset$ then we say that the tissue P system is *without environment*.

A symport rule $(i, u/\lambda, j)$, with $i \neq 0, j \neq 0$, provides a virtual arc from cell i to cell j. An antiport rule $(i,\ u\ /\ v,\ j)$, with $i \neq 0, j \neq 0$, provides two arcs: one from cell i to cell j and another one from cell j to cell i. Thus, every tissue P systems has an underlying directed graph whose nodes are the cells of the system and the arcs are obtained from communication rules. In this context, the environment can be considered as a virtual node of the graph such that their connections are defined by the communication rules of the form $(i,\ u\ /\ v,\ j)$, with $i = 0$ or $j = 0$.

2.2 Semantics

An *instantaneous description* or *configuration* $\mathcal{C}_t$ at an instant t of a tissue P system with symport/antiport rules is described by all multisets of objects over Γ associated with all the cells present in the system, and the multiset of objects over $\Gamma \backslash \mathcal{E}$ 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 Π is $(\mathcal{M}_1, \ldots, \mathcal{M}_q; \emptyset)$.

A symport rule $(i, u/\lambda, j)$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if the following holds: (a) there exists a zone labelled by i and other zone labelled by j at configuration $\mathcal{C}_t$; and (b) multiset u is contained in such zone i. When applying a rule $(i, u/\lambda, j)$, the objects specified by multiset u are sent to zone j.

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

The rules of a tissue 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 rules can be added.

Given a tissue P system with symport/antiport rules Π, we say that configuration $\mathcal{C}_t$ yields configuration $\mathcal{C}_{t+1}$ when the rules from $\mathcal{R}$ are applied following the previous remarks.

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 manner; and (c) if the sequence is finite (called *halting computation*) then the last term of the sequence is a *halting configuration*.

2.3 Cell division and cell separation

Here, we introduce new types of rules (cell division and cell 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 cells) in linear time.

Definition 2 A tissue P system with symport/antiport rules and cell division of degree $q \geq 1$ is a tuple $\Pi = (\Gamma, \mathcal{E}, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$, where:

1. $\Pi = (\Gamma, \mathcal{E}, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$ is a tissue P system with symport/antiport rules;
2. The set $\mathcal{R}$ also contains rules of the form $[\, a \,]_i \rightarrow [\, b \,]_i [\, c \,]_i$, where $1 \leq i \leq q, i \neq i_{out}$ and $a, b, c \in \Gamma$ (division rules).

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

Definition 3 A tissue P system with symport/antiport rules and cell separation of degree $q \geq 1$ is a tuple $\Pi = (\Gamma, \Gamma_0, \Gamma_1, \mathcal{E}, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$, where:

1. $\Pi = (\Gamma, \mathcal{E}, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$ is a tissue 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. The set $\mathcal{R}$ also contains rules of the form $[\, a \,]_i \rightarrow [\, \Gamma_0 \,]_i [\, \Gamma_1 \,]_i$, where $1 \leq i \leq q, i \neq i_{out}$ and $a \in \Gamma$ (separation rules).

A separation rule $[\, a \,]_i \rightarrow [\, \Gamma_0 \,]_i [\, \Gamma_1 \,]_i \in \mathcal{R}$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if there exists a cell labelled by i at configuration $\mathcal{C}_t$ and object a is contained in such cell. When applying a separation rule $[\, a \,]_i \rightarrow [\, \Gamma_0 \,]_i [\, \Gamma_1 \,]_i$ to such cell i, under the influence of object a, the cell with label i is separated into two new cells with the same label; at the same time, object a is consumed; the objects from Γ_0 are placed in the first cell and those from Γ_1 are placed in the second cell.

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 cell i is divided (resp., separated), the division rule (resp., separation rule) is the only one from $\mathcal{R}$ which is applied for that cell at that step. The new cells resulting from division (resp., separation) could participate in the interaction with other cells or the environment by means of communication rules at the next step—providing that they are not divided (resp., separated) once again.

3 Cell-like P systems with symport/antiport rules

Cell-like P systems, or simply P systems were initially though as rewriting systems working in a hierachical structure of membranes and depending on the membrane that an object is placed, it will evolve in one way or another one. Later in [11], taking the idea of symport/antiport rules from tissue P systems, a new framework was developed to compare the frontiers of efficiency and see if the structure of the graph associated with the system matters in terms of problems that can be efficiently solved.

3.1 Syntax

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. $\mathcal{E} \subsetneq \Gamma$;
3. μ is a rooted tree whose nodes are injectively labelled with $1, \ldots, q$;
4. $\mathcal{M}_1, \ldots, \mathcal{M}_q$ are multisets over Γ;
5. $\mathcal{R} = \mathcal{R}_1 \cup \ldots \cup \mathcal{R}_q$, where $\mathcal{R}_i$ is a finite set of communication rules associated with node i, of the following forms:

   - (u, *out*) or (u, *in*), where u is a non-empty multiset over Γ (symport rules);
   - (u, *out*; v, *in*), where u, v are non-empty multisets over Γ (antiport rules);

6. $i_{out} \in \{0, 1, 2, \ldots, q\}$.

A P system with symport/antiport rules $\Pi = (\Gamma, \mathcal{E}, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$ of degree $q \geq 1$ can be viewed as a set of q membranes labelled by $1, \ldots, q$, arranged in a hierarchical structure μ given by a rooted tree whose root is called the *skin membrane* labelled by 1, such that: (a) $\mathcal{M}_1, \ldots, \mathcal{M}_q$ are multisets over the *working alphabet* 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) $\mathcal{R}_1, \ldots, \mathcal{R}_q$ are finite sets of communication rules over Γ ($\mathcal{R}_i$ is associated with the membrane i of μ); and (d) i_{out} 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 $i = 0$. The length of rule (u, *out*) or (u, *in*) (resp., (u; *out*; v, *in*)) is defined as $\mid u \mid$ (resp., $\mid u \mid + \mid v \mid$). If $\mathcal{E} = \emptyset$ then we say that the P system is *without environment*.

For each membrane i 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.

3.2 Semantics

An *instantaneous description* or *configuration* $\mathcal{C}_t$ 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 $\Gamma \setminus \mathcal{E}$ 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 $(\mu, \mathcal{M}_1, \ldots, \mathcal{M}_q; \emptyset)$.

A symport rule $(u, out) \in \mathcal{R}_i$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if there exists a membrane labelled by i at configuration $\mathcal{C}_t$, and multiset u is contained in such membrane. When applying a rule $(u, out) \in \mathcal{R}_i$ to such membrane i, the objects specified by multiset u are sent to the zone inmediately outside of such membrane, that is, to zone $p(i)$.

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

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

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 rules can be added.

Given a P system with symport/antiport rules Π, we say that configuration $\mathcal{C}_t$ yields configuration $\mathcal{C}_{t+1}$ by applying the rules from $\mathcal{R}$ following the previous remarks. The concept of *computation* is analogous to the one defined for tissue P systems with symport/antiport rules.

3.3 Membrane division and membrane separation

Here, 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. $\mathcal{R} = \mathcal{R}_1 \cup \ldots \cup \mathcal{R}_q$, where for each $i \notin \{1, i_{out}\}$ such that i is the label of an elementary membrane, $\mathcal{R}_i$ also contains rules of the form $[\,a\,]_i \rightarrow [\,b\,]_i [\,c\,]_i$, where $a, b, c \in \Gamma$ (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 there exists an elementary membrane labelled by i at configuration $\mathcal{C}_t$, and object a is contained in such membrane. When applying a division rule $[\,a\,]_i \rightarrow [\,b\,]_i[\,c\,]_i$ to an elementary membrane labelled by i at configuration $\mathcal{C}_t$, under the influence of object a, such membrane is divided into two new membranes with the same label; in the first copy, object a is replaced by object b and 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. $\mathcal{R} = \mathcal{R}_1 \cup \ldots \cup \mathcal{R}_q$, where for each $i \notin \{1, i_{out}\}$ such that i is the label of an elementary membrane, $\mathcal{R}_i$ also contains rules of the form $[\,a\,]_i \rightarrow [\,\Gamma_0\,]_i[\,\Gamma_1\,]_i$, where $a \in \Gamma$ (separation rules for elementary membranes).

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 there exists an elementary membrane labelled by i at configuration $\mathcal{C}_t$, and object a is contained in such membrane. When applying a separation rule $[\,a\,]_i \rightarrow [\,\Gamma_0\,]_i[\,\Gamma_1\,]_i$ to an elementary membrane labelled by i, under the influence of object a, such membrane is separated into two new membranes with the same label; at the same time, object a is consumed; the objects from Γ_0 are placed in the first membrane and 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. 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.

4 Solving decision problems by means of membrane systems

Originally called *accepting* membrane systems in [16], *recognizer* membrane systems are devices devoted to solve decision problems. It is worth noting that solving decision problems can be described by recognizing a certain language associated with it.

4.1 Recognizer membrane systems

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

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

where we can think that $\Gamma_0 = \Gamma_1 = \emptyset$ for membrane systems without separation rules and μ is not explicitely defined in tissue P systems.

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

Definition 7 A recognizer membrane system

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

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

We denote by $\mathcal{TC}$ (respectively, $\mathcal{TDC}$, $\mathcal{TSC}$) the class of recognizer tissue P systems with symport/antiport rules (resp., with division rules or separation rules). For each natural number $k \geq 1$, we denote by $\mathcal{TC}(k)$ (resp., $\mathcal{TDC}(k)$, $\mathcal{TSC}(k)$) the class of recognizer tissue P systems with symport/antiport rules (resp., with division rules or with separation rules) such that the length of the communication rules is at most k. If the set associated with the environment is empty then we write $\widehat{\mathcal{TC}}$, $\widehat{\mathcal{TDC}}$, $\widehat{\mathcal{TSC}}$, $\widehat{\mathcal{TC}}(k)$, $\widehat{\mathcal{TDC}}(k)$, $\widehat{\mathcal{TSC}}(k)$, respectively.

We denote by $\mathcal{CC}$ (respectively, $\mathcal{CDC}$, $\mathcal{CSC}$) the class of recognizer P systems with symport/antiport rules (resp., with division rules or separation rules). For each natural number $k \geq 1$, we denote by $\mathcal{CC}(k)$ (respectively, $\mathcal{CDC}(k)$, $\mathcal{CSC}(k)$) the class of recognizer P systems with symport/antiport rules (resp., with division rules or with separation rules) such that the length of the communication rules is at most k. If the set associated with the environment is empty then we write $\widehat{\mathcal{CC}}$, $\widehat{\mathcal{CDC}}$, $\widehat{\mathcal{CSC}}$, $\widehat{\mathcal{CC}}(k)$, $\widehat{\mathcal{CDC}}(k)$, $\widehat{\mathcal{CSC}}(k)$, respectively.

4.2 Polynomial complexity classes of recognizer membrane systems

According to [16], we define what solving a decision problem by a family of recognizer membrane 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 $\Pi = \{\Pi(n) \mid n \in \mathbb{N}\}$ of recognizer membrane systems (in a uniform way) if the following hold:

- the family Π 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 Π 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 post $p(\mid u \mid)$ steps;
  - the family Π 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 Π 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 this definition, 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 [16], for details).

5 Efficiency of computing models

Let us recall that each computing model provided a mathematical definition of the informal idea of solving abstract problemas by means of mechanical procedure (*algorithm*). A computing model which is equivalent in power to Turing machines is called *universal*. An algorithm in a universal computing model is *efficient* if it runs in polynomial time.

An abstract problem is called *tractable* if it can be solved by an efficient algorithm in a universal computing model, that is, if it is solvable by a polynomial-time algorithm (the upper bound of computational resources is polynomial). The complexity class of decision tractable problems is denoted by **P**. An abstract problem is called *intractable* if it cannot be solved by a polynomial-time algorithm (the lower bound computational resources is exponential).

Let us recall that **NP**-complete problems are problems with no known polynomial-time algorithm but not yet proven to be intractable. They are merely conjectured to be so (assuming that $\mathbf{P} \neq \mathbf{NP}$) and we say that **NP**-complete problems are *presumably intractable*. A computing model with the ability to provide polynomial-time solutions to intractable problems (resp., **NP**-complete problems) is called an *efficient computing model* (resp., *presumably efficient computing model*).

Given two computing models M_1 and M_2, we say that M_1 is a *submodel* of M_2, denoted by $M_1 \subseteq M_2$, if and only if each mechanical procedure in M_1 is also a mechanical procedure in M_2. Thus, if $M_1 \subseteq M_2$ then model M_2 is obtained from model M_1 by adding some syntactic or semantic ingredients, and each mechanical solution S in M_1 to an abstract problem X is also a mechanical solution S in M_2. Let us assume that M_1 and M_2 are computing models such that: (a) M_1 is non-efficient; (b) M_2 is efficient; and (c) $M_1 \subseteq M_2$. Then, we can think that passing from computing model M_2 to computing model M_1 amounts to passing from non-efficiency to efficiency. In this context, we also say that ingredients added to M_1 to produce M_2, provides a frontier of the efficiency, that is, a borderline of the tractability of abstract problems.

Let S_X be a polynomial-time solution to an **NP**-complete problem X in a presumably efficient computing model M_2. In order to prove that $\mathbf{P} \neq \mathbf{NP}$ would be enough to show that removing the ingredients needed to obtain M_2 from M_1 is not possible to produce a polynomial-time solution to X. In order to prove that $\mathbf{P} = \mathbf{NP}$ would be enough to generate from S_X a new polynomial-time solution to X such that

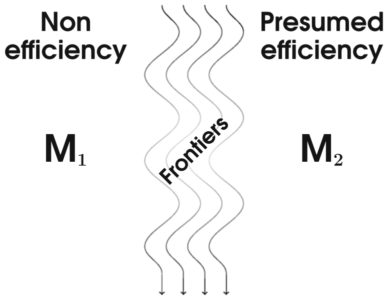

Fig. 1 A new way to tackle the **P** vs. **NP**problem

does not use the ingredients needed to obtain M_2 from M_1. A grpahical representation of this can be seen in Fig. 1

Let us recall that the classical approach to tackle the the **P** versus **NP** problem is to consider it an **NP**-complete problem and try to prove that it is a problem in class **P**. If this is possible then $\mathbf{P} = \mathbf{NP}$, otherwise $\mathbf{P} \neq \mathbf{NP}$. According with the previous considerations, each frontier of the efficiency provides a new and unconventional way to attack the **P** versus **NP** problem.

5.1 Techniques

In order to obtain results that provide frontiers of the efficiency in the framework of Membrane Computing, three important techniques (dependency graph technique, simulation technique and algorithmic technique) have been used. Next, these techniques are briefly described.

5.1.1 Dependency graph technique

Let Π be a recognizer tissue P system where all its communication rules have length 1. In this case, each rule of Π can be activated by a single object (note that this holds also for division or separation rules). Hence, there exists in some sense, a *dependency* between the object triggering the rule and the object or objects produced by its application. Then, a directed graph (*dependency graph*) can be associated with Π verifying the following relevant property: 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 (see [17] and [18] for more details).

5.1.2 Simulation technique

Let us define the meaning of efficient simulations in the framework of recognizer tissue P systems. Given two recognizer tissue P systems, Π and Π', we say that Π' *simulates* Π *in an efficient way* if the following holds: (a) Π' can be constructed from Π by a deterministic Turing machine working in polynomial time; and (b) There exists an injective function, f, from the set $\mathbf{Comp}(\Pi)$ of computations of Π onto the set $\mathbf{Comp}(\Pi')$ of computations of Π' such that:

- ★ There exists a deterministic Turing machine that constructs computation $f(\mathcal{C})$ from computation $\mathcal{C}$ in polynomial time.
- ★ A computation $\mathcal{C} \in$ **Comp** (Π) is an accepting computation if and only if $f(\mathcal{C}) \in$ **Comp** (Π') is an accepting one.
- ★ There exists a polynomial function $p(n)$ such that for each $\mathcal{C} \in$ **Comp** (Π) we have $|f(\mathcal{C})| \leq p(|\mathcal{C}|)$.

5.1.3 Algorithmic technique

The technique consists of the construction of a deterministic algorithm $\mathcal{A}$ working in polynomial time that receives as input a tissue P system Π from $\mathcal{F}$ and an input multiset m of Π. Then, algorithm $\mathcal{A}$ reproduces the behaviour of a computation of $\Pi + m$. In particular, if the given tissue P system is confluent then the algorithm will provide the same answer of the system, that is, the answer of algorithm $\mathcal{A}$ is affirmative if and only if the system $\Pi + m$ has an accepting computation (and then, any computation is an accepting one).

6 On efficiency of membrane systems with symport/antiport rules

In this section we analyze the computational efficiency of recognizer membrane systems (tissue-like P systems or cell-like P systems) with symport/antiport rules.

6.1 Basic membrane systems with symport/antiport rules

We analyze the limits on efficient computation of basic membrane systems (tissue-like P systems or cell-like P systems) with symport/antiport rules.

It is worth noting that in this kind of tissue P systems, an exponential workspace (expressed in terms of the number of objects) can be constructed in linear time. Indeed, let us consider a cell-like or a tissue-like P system of degree 1 whose set of the environment is $\mathcal{E} = \{a_1, \ldots, a_{n+1}\}$, with $n \geq 1$, and the finite set of rules is $\mathcal{R} = \{(1, a_i/a_{i+1}^2, 0) \mid 1 \leq i \leq n\}$. Then 2^n objects a_{n+1} can be produced in n computation steps. However, this property is not enough in order to efficiently solve computationally hard problems. Indeed, on the one hand, families of recognizer basic tissue P systems which solve problem can be efficiently simulated by a family of recognizer basic transition P systems solving the same problem (see [19], for details). On the other hand, it is well known that only problems in class **P** can be solved in polynomial time by means of families of recognizer basic transition P systems [20]. Then we have the following result:

Proposition 1 $\mathbf{P} = \mathbf{PMC}_{\mathcal{TC}} = \mathbf{PMC}_{\mathcal{CC}}$.

6.2 Membrane systems with symport/antiport rules and division rules

Allowing the use of division rules to create an exponential workspace in terms of cells or membranes seems powerful. Here we analyze the efficiency of recognizer tissue-like and cell-like P systems with symport/antiport rules from a computational complexity point of view. Specifically, the ability to solve computationally hard problems by means of families of such recognizer membrane systems is studied.

By using the technique of dependency graph associated with tissue-like and cell-like P systems with cell division and communication rules with length at most 1, it has been proved that this kind of membrane systems can only efficiently solve problems in class **P** (see [17] and [21], for details).

Proposition 2 $\mathbf{P} = \mathbf{PMC}_{\mathcal{TDC}(1)} \cap \mathbf{PMC}_{\mathcal{CDC}(1)}$

On the one hand, in [22], a polynomial-time solution of the HAM-CYCLE problem, a well known **NP**-complete problem, was given by using a family of recognizer tissue P systems with cell division and communication rules of length at most 2. On the other hand, in [23], a polynomial-time solution of the HAM-CYCLE problem, was given by using a family of recognizer P systems with membrane division division and communication rules of length at most 2. Therefore, we have:

Proposition 3 $\mathbf{NP} \cup \textbf{co-}\ \mathbf{NP} \subseteq \mathbf{PMC}_{\mathcal{TDC}(2)} \cap \mathbf{PMC}_{\mathcal{CDC}(2)}$.

6.3 Membrane systems with symport/antiport rules and separation rules

By using the simulation technique, it has been proved that only problems in class **P** can be solved in polynomial time by means of families of tissue-like (resp. cell-like) P systems with cell separation (resp. membrane separation) which use communication rules with length at most 2 (see [24] for tissue-like P systems and [14] for cell-like P systems).

Proposition 4 $\mathbf{P} = \mathbf{PMC}_{\mathcal{TSC}(2)} \cap \mathbf{PMC}_{\mathcal{CSC}(2)}$

On the one hand, in [15], a polynomial-time solution of the SAT problem was given by using a family of recognizer tissue P systems with cell division and communication rules of length at most 3. On the other hand, in [14], a polynomial-time solution of the SAT problem, was given by using a family of recognizer P systems with membrane separation and communication rules of length at most 3. Therefore, we have:

Proposition 5 $\mathbf{NP} \cup \textbf{co-}\ \mathbf{NP} \subseteq \mathbf{PMC}_{\mathcal{TSC}(3)} \cap \mathbf{PMC}_{\mathcal{CSC}(3)}$.

6.4 Membrane systems with symport/antiport rules and without environment

By using the algorithmic technique, it has been proved that only problems in class **P** can be solved in polynomial time by means of families of tissue-like P systems with symport/antiport rules and cell separation but without environment (see [25] for tissue-like P systems and [26] for cell-like P systems). Therefore, we have:

Proposition 6 $\mathbf{P} = \mathbf{PMC}_{\widehat{\mathcal{TSC}}} = \mathbf{PMC}_{\widehat{\mathcal{CSC}}}$

By using the simulation technique, it has been proved that each family of recognizer tissue-like (resp., cell-like) P systems with cell division which use communication rules of length at most $k \geq 1$ and solve a decision problem X in polynomial time, can be efficiently simulated by means of a family of recognizer cell-like P systems with cell division and without environment which use communication rules of length at most $k \geq 1$, solving X in polynomial time (see [27] for cell-like P systems, and [28] for cell-like P systems). Therefore,

Proposition 7 For each $k \geq 1$ we have:
$\mathbf{PMC}_{\widehat{\mathcal{TDC}}(k)} = \mathbf{PMC}_{\mathcal{TDC}(k)} = \mathbf{PMC}_{\widehat{\mathcal{CDC}}(k)} = \mathbf{PMC}_{\mathcal{CDC}(k)}$

7 New frontiers of the tractability

In this section, new frontiers of the tractability in terms of membrane systems with symport/antiport rules, are obtained.

- From Propositions 1 and 3, a frontier of the tractability is obtained when the use of cell division rules (resp. membrane division rules) are allowed in basic tissue P systems (resp. cell-like P systems) with symport/antiport rules.
- From Propositions 2 and 3, we deduce that in the framework of recognizer tissue P systems with cell division, the length of the communication rules provides a borderline of the tractability of decision problems. Specifically, passing from length 1 to length 2, amounts to passing from non-efficiency to efficiency, assuming that $\mathbf{P} \neq \mathbf{NP}$.
- From Propositions 4 and 5, we deduce that in the framework of recognizer tissue-like (resp., cell-like) P systems with cell separation (resp., membrane separation), the length of the communication rules provides a borderline of the tractability of decision problems.

Specifically, passing from length 2 to length 3, amounts to passing from non-efficiency to efficiency, assuming that $\mathbf{P} \neq \mathbf{NP}$.

- From Propositions 5 and 6, we deduce that in the framework of recognizer tissue-like (resp., cell-like) P systems with cell separation (resp., membrane separation), and the communication rules with length at most 3, the environment provides a borderline of the tractability of decision problems. Specifically, passing from don't have environment to have environment, amounts to passing from non-efficiency to efficiency, assuming that $\mathbf{P} \neq \mathbf{NP}$.
- From Proposition 7, we deduce that in the framework of recognizer tissue-like (resp., cell-like) P systems with symport/antiport rules and cell division (resp., membrane division), the role of the environment is irrelevant from a computational complexity point of view.

8 Conclusions and future work

Membrane systems (tissue-like and cell-like) with symport/antiport rules, with or without environment, which use division rules (inspired by the mitosis) or separation rules (inspired to membrane fission) to implement a mechanism able to produce an exponential workspace (expressed in terms of number of objects and number of cells/membranes) in poynomial-time, has been analyzed from a computational complexity point of view.

Assuming that $\mathbf{P} \neq \mathbf{NP}$, some frontiers of the efficiency have been obtained, providing a new (unconventional computing) technique to address the **P** versus **NP** problem. In this context, has been shown that the structure (a *directed graph* in the tissue-like approach and a *rooted tree* in the case of cell-like approach) is not relevant. It seems interesting to study if the structure matters when working with active membranes in cell-like P systems or with active cells in tissue-like P systems. Another relevant research line is the fact that **PSPACE**-complete problems as $\texttt{QBF} - \texttt{SAT}$ can be efficiently solved by means of a family of cell-like P systems with symport/antiport with length at most 3 and division rules [29]. Here, the hierarchized structure formed by the rules of the system helps in the simulation of the quantifiers. On the contrary, in [30] it has been demonstrated that the upper bound of tissue-like P systems with symport/antiport rules and division rules is the complexity class $\mathbf{P}^{\#\mathbf{P}}$, obtaining a frontier between these two complexity classes from the Membrane Computing framework. Several families of membrane systems have the exact characterization of **P**, but it is not usual to have the equivalent complexity class if the family can solve presumably hard problems, so another interesting research topic is to characterize existing membrane computing families.

Acknowledgements This work was partially supported by Project TIN2017-89842-P, cofinanced by Ministerio de Economía, Industria y Competitividad (MINECO) of Spain, through the Agencia Estatal de Investigación (AEI), and by Fondo Europeo Regional (FEDER) of the European Union and by Grant number 61320106005 of the National Natural Science Foundation of China.

References

1. Păun, G.: Computing with membranes. J. Comput. Syst. Sci. **61**(1), 108–143 (2000)
2. Ionescu, M., Păun, G., Yokomori, T.: Spiking neural P systems. Fundam. Inf. **71**(2–3), 279–308 (2006)
3. Martín-Vide, C., Păun, G., Pazos, J., Rodríguez-Patón, A.: Tissue P systems. Theor. Comput. Sci. **296**(2), 295–326 (2003)
4. Păun, G.: Membrane Computing: An Introduction. Springer Natural Computing Series (2002)
5. Paun, Gh., Rozenberg, G., Salomaa, A.: The Oxford Handbook of Membrane Computing. Oxford University Press, Oxford, UK (2009)
6. Frisco, P., Gheorghe, M., Pérez-Jiménez, M.J. (eds.): Applications of Membrane Computing in Systems and Synthetic Biology. Springer, Cham, Switzerland (2014)
7. Pan, L., Paun, Gh., Pérez-Jiménez, M.J., Song, T.: Bio-inspired Computing: Theories and Applications. Communications in Computer and Information Science Series (2014)
8. Zhang, G., Pérez-Jiménez, M.J., Gheorghe, M.: Real-life applications with Membrane Computing. Emergence, Complexity and Computation Series (2017)
9. Lodish, H., Berk, A., Zipursky, S.L., Matsudaira, P., Baltimore, D., Darnell, J.: Biomembranes: Structural Organization and Basic Functions. Molecular Cell Biology, 4th edn. W. H Freeman, New York (2000)
10. Padan, E., Landau, M.: Sodium–Proton (Na+/H+) Antiporters: properties and roles in health and disease. In: Sigel, A., Sigel, H., Sigel, R.K.O. (eds.) The Alkali Metal Ions: Their Role in Life. Metal Ions in Life Sciences, vol. 16, pp. 391–458. Springer, Berlin (2016)
11. Păun, A., Păun, G.: The power of communication: P systems with symport/antiport. New Gener. Comput. **20**, 295–305 (2002)
12. Macías-Ramos, L.F., Song, B., Song, T., Pan, L., Pérez-Jiménez, M.J.: On efficiency of P systems with symport/antiport and membrane division. In: Proceedings of the Fifteenth Brainstorming Week on Membrane Computing, Fénix Editora, pp. 147–160. Sevilla, Spain, 31 January–3 February 2017
13. Păun, G., Pérez-Jiménez, M.J., Riscos-Núñez, A.: Tissue P systems with cell division. Int. J. Comput. Commun. **3**, 295–303 (2008)
14. Macías-Ramos, L.F., Song, B., Valencia-Cabrera, L., Pan, L., Pérez-Jiménez, M.J.: Membrane fission: a computational complexity perspective. Complexity **21**(6), 321–334 (2016)
15. Pérez-Jiménez, M.J., Sosík, P.: An optimal frontier of the efficiency of tissue P systems with cell separation. Fundam. Inf. **138**(1–2), 45–60 (2015)
16. Pérez-Jiménez, M.J., Romero-Jiménez, A., Sancho-Caparrini, F.: Complexity classes in models of cellular computing with membranes. Nat. Comput. **2**(3), 265–285 (2003)

17. Gutiérrez-Escudero, R., Pérez-Jiménez, M.J., Rius-Font, M.: Characterizing tractability by tissue-like P systems. In: Lecture Notes in Computer Science, vol. 5957, pp. 289–300 (2010)
18. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A., Romero-Campero, F.J.: On the power of dissolution in P systems with active membranes. In: Lecture Notes in Computer Science, vol. 3850, pp. 224–240 (2006)
19. Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Romero-Jiménez, A.: Efficient simulation of tissue-like P systems by transition cell-like P systems. Nat. Comput. **8**(4), 797–806 (2009)
20. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A., Romero-Campero, F.J., Romero-Jiménez, A.: Characterizing tractability by cell-like membrane systems. In: Subramanian, K.G., Rangarajan, K., Mukund, M. (eds.) Formal Models, Languages and Applications, pp. 137–154. World Scientific, Singapore (2006)
21. Macías-Ramos, L.F., Song, B., Song, T., Pan, L., Pérez-Jiménez, M.J.: Limits on efficient computation in P systems with symport/antiport. In: Graciani, C., Păun, G., Riscos-Núñez, A., Valencia-Cabrera, L. (eds.) Proceedings of the Fifteenth Brainstorming Week on Membrane Computing, pp. 147–160. Sevilla, Spain, 31 January–3 February 2017
22. Porreca, A.E., Murphy, N., Pérez-Jiménez, M.J.: An optimal frontier of the efficiency of tissue P systems with cell division. In: Garcí-Quismondo, M., Macías-Ramos, L.F., Păun, G., Pérez Hurtado, I., Valencia-Cabrera, L. (eds.) Proceedings of the Tenth Brainstorming Week on Membrane Computing, vol. II, Report RGNC 01/2012, Fénix Editora, pp. 141–166. Seville, Spain, 30 January–3 February, 2012
23. Valencia-Cabrera, L., Song, B., Macías-Ramos, L.F. , Pan, L., Riscos-Núñez, A., Pérez-Jiménez, M.J.: Minimal cooperation in P systems with symport/antiport: A complexity approach. In: Macías, L.F., Păun, G., Riscos, A., Valencia, L. (eds) Proceedings of the Thirteenth Brainstorming Week on Membrane Computing, pp. 301–323. Sevilla, Spain, 2–6 February 2015
24. Pan, L., Pérez-Jiménez, M.J., Riscos-Núñez, A., Rius, M.: New frontiers of the efficiency in tissue P systems. In: Pan, L., Păun, G., Song, T. (eds.) Pre-proceedings of Asian Conference on Membrane Computing (ACMC 2012), Huazhong University of Science and Technology, pp. 61–73. Wuhan, China, 15–18 October 2012
25. Macías-Ramos, L.F., Pérez-Jiménez, M.J., Riscos-Núñez, A., Rius, M., Valencia-Cabrera, L.: The efficiency of tissue P systems with cell separation relies on the environment. In: Lecture Notes in Computer Science, vol. 7762, pp. 243–256 (2013)
26. Macías-Ramos, L.F., Pérez-Jiménez, M.J., Riscos-Núñez, A., Valencia-Cabrera, L.: Membrane fission versus cell division: when membrane proliferation is not enough. Theor. Compu. Sci. **608**, 57–65 (2015)
27. Pérez-Jiménez, M.J., Riscos-Núñez, A., Rius, M., Romero, F.J. : The role of the environment in tissue like P systems with cell division. In: García-Quismondo, M., Macías-Ramos, L.F., Păun, G., Pérez-Hurtado, I., Valencia-Cabrera, L. (eds.) Proceedings of the Tenth Brainstorming Week on Membrane Computing, vol. II, Report RGNC 01/2012, Fénix Editora, pp. 89–104. Seville, Spain, 30 January–3 February 2012
28. Macías-Ramos, L.F., Song, B., Song, T., Pan, L., Pérez-Jiménez, M.J.: On efficiency of P systems with symport/antiport and membrane division. Int. J. Inf. Theor. Appl. **24**(3), 229–245 (2017)
29. Song, B., Pérez-Jiménez, M.J., Pan, L.: Efficient solutions to hard computational problems by P systems with symport/antiport rules and membrane division. Biosystems **130**, 51–58 (2015)
30. Leporati, A., Manzoni, L., Mauri, G., Porreca, A.E., Zandron, C.: Characterising the complexity of tissue P systems with fission rules. J. Comput. Syst. Sci. **90**, 115–128 (2017)