Journal of Automata, Languages and Combinatorics **21** (2016) 1–2, 107 – 123

POLARIZATIONLESS P SYSTEMS WITH ACTIVE MEMBRANES: COMPUTATIONAL COMPLEXITY ASPECTS

Luis Valencia-Cabrera David Orellana-Martín
Miguel A. Martínez-del-Amor Agustín Riscos-Núñez
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
{lvalencia, dorellana, mdelamor, ariscosn, marper}@us.es

ABSTRACT

P systems with active membranes, in their classical definition, make use of non-cooperative rules only. However, it is well known that in living cells, proteins interact among them yielding new products. Inspired by this biological phenomenon, the previous framework is reformulated in this paper, allowing cooperation in object evolution rules, while removing electrical charges associated with membranes. More precisely, minimal cooperation in object evolution rules is incorporated in polarizationless P systems with active membranes. In this paper, the term "minimal" means that the left-hand side of such rules consists of at most two symbols, and its length is greater than or equal to the corresponding right-hand side. The computational efficiency of this kind of P systems is studied by providing a uniform polynomial-time solution to SAT problem in such manner that only division rules for elementary membranes are used and dissolution rules are forbidden. Bearing in mind that only tractable problems can be efficiently solved by families of polarizationless P systems with active membranes and without dissolution rules, passing from non-cooperation to minimal cooperation in object evolution rules amounts passing from non-efficiency to efficiency in this framework. This frontier of efficiency provides, as any other borderline does, a possible way to address the **P** versus **NP** problem.

Keywords: membrane computing, active membranes, minimal cooperation, mitosis, computational complexity, the **P** versus **NP** problem

1. Introduction

A possible way of producing new membranes in living cells is based on the *mitosis* process. *Mitosis* is a basic process of each cell life cycle in eukaryotic cells which allows producing two or more cells from one cell that could be considered as the "mother". Several mechanisms based on cell division were introduced in *Membrane Computing* [11], a distributed parallel computing paradigm inspired by the way the living cells process chemical substances, energy and information. The processor units

in the basic model are abstractions of biological membranes, selectively permeable barriers which give cells their outer boundaries (plasma membranes) and their inner compartments (organelles). They control the flow of information between cells, the movement of substances into and out of cells, and they are also involved in the capture and release of energy. Biological membranes play an active part in the life of the cell. In fact, the passing of chemical substances through a biological membrane often requires an interaction between the membrane itself and the protein channels present within it. During this interaction, both the chemical substances and the membrane itself can be modified at least locally.

P systems with active membranes [8] incorporate the mitosis based mechanisms by means of *membrane division* rules. By applying this kind of rules, under the influence of the object triggering it, the membrane is divided into two membranes and that object is replaced in the two new ones by possibly different objects, being the remaining objects *duplicated* in both the created membranes. These models are computationally universal (they are equivalent in power to deterministic Turing machines) and they have the ability to provide efficient solutions to computationally hard problems, by making use of an exponential workspace created in a polynomial time (often, in linear time). Moreover, **PSPACE**-complete problems can be efficiently solved by families of P systems with active membranes which use division for elementary and non-elementary membranes. Consequently, the usual framework of P systems with active membranes and polarizations for solving decision problems seems to be too powerful from the computational complexity point of view.

P systems with active membranes and without electrical charges were initially studied in [1, 2]. However, this initial approach proposed to replace polarizations by a somehow equivalent or even more powerful ingredient: the ability to change the label of the membranes. Our approach is quite different, since we keep labels unchanged. Bearing in mind that consideration, some computational complexity aspects of these systems are studied when (minimal) cooperation in object evolution rules is allowed, and a new frontier of the efficiency is provided.

The paper is organized as follows. Next section briefly describes some preliminaries in order to make the work self-contained. In Section 3, polarizationless P systems with active membranes are introduced, and minimal cooperation in object evolution rules is considered. A uniform polynomial time solution to `SAT` problem by means of a family of polarizationless P systems with active membranes, with minimal cooperation and without dissolution rules, is presented in Section 4. The paper ends with some open problems and concluding remarks.

2. Preliminaries

An *alphabet* Γ is a non-empty set and its 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 $|u|$, that is, the length of a string is the number of occurrences of symbols it contains. 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 (respectively, empty) if its support is a finite (respectively, empty) set. We denote by $\emptyset$ the empty multiset and we denote by $M_f(\Gamma)$ the set of all finite multisets over Γ. If $m_1 = (\Gamma, f_1)$ and $m_2 = (\Gamma, f_2)$ are 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 $x \in \Gamma$.

Let us recall that 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 (y, x) (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* (see [4] for details).

Finally, let us recall that a *decision problem* X is a pair (I_X, Θ_X) where I_X is a language over a finite alphabet and Θ_X is a Boolean formula over the set I_X (called the set of instances of X).

3. Polarizationless P Systems with Active Membranes

Let us briefly recall some definitions of P systems models that will be used in the paper (see [11] for details). The reader is supposed to be familiar with basic elements of membrane computing.

A *basic transition* P system is a membrane system whose rules are of the following forms: evolution, communication, and dissolution. In these systems the size of the membrane structure does not increase, but an exponential workspace (in terms of number of objects) can be constructed in linear time, e. g. via evolution rules of the type $[\, a \to a^2 \,]_h$. Nevertheless, this capability is not enough to efficiently solve **NP**–complete problems, unless $\mathbf{P} = \mathbf{NP}$ (see [7] for details).

Replication is one of the most important functions of a cell and, in ideal circumstances, a cell produces two identical copies by division. Bearing in mind that the reactions which take place in a cell are related to membranes, division rules for elementary and non-elementary membranes are considered in the so-called *P systems with active membranes*. This variant was first introduced by Gh. Păun [9] and it has electrical charges associated with membranes, but the rules are non-cooperative and there are not priorities. Nevertheless, the class of all problems solvable in polynomial time in a uniform way by means of families of P systems with active membranes which use division for elementary and non-elementary membranes coincides with **PSPACE** [15].

3.1. Polarizationless P systems with active membranes: Syntax

In this paper, electrical charges are removed, so we deal with polarizationless P systems with active membranes.

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

- Γ is a finite alphabet whose elements are called objects;
- H is a finite alphabet such that $H \cap \Gamma = \emptyset$ whose elements are called labels;
- μ is a labelled rooted tree (called membrane structure) consisting of q nodes injectively labeled by elements of H;
- $\mathcal{M}_1, \ldots, \mathcal{M}_q$ are multisets over Γ;
- $\mathcal{R}$ is a finite set of rules, of the following forms:
  - (a_0) $[\, a \to u \,]_h$ for $h \in H$, $a \in \Gamma$, $u \in M_f(\Gamma)$ *(object evolution rules).*
  - (b_0) $a[\ \ ]_h \to [\, b]_h$ for $h \in H$, $a, b \in \Gamma$ and h is not the label of the root of μ *(send–in communication rules).*
  - (c_0) $[\, a\,]_h \to b\,[\ \ ]_h$ for $h \in H$, $a, b \in \Gamma$ *(send–out communication rules).*
  - (d_0) $[\, a\,]_h \to b$ for $h \in H \setminus \{i_{out}\}$, $a, b \in \Gamma$ and h is not the label of the root of μ *(dissolution rules).*
  - (e_0) $[\, a\,]_h \to [\, b\,]_h\, [\, c\,]_h$ for $h \in H \setminus \{i_{out}\}$, $a, b, c \in \Gamma$ and h is the label of an elementary membrane different of the root of μ *(division rules for elementary membranes).*
  - (f_0) $[\,[\,]_{h_1}[\,]_{h_2}\,]_{h_0} \to [\,[\,]_{h_1}\,]_{h_0}\, [\,[\,]_{h_2}\,]_{h_0}$, where $h_0, h_1, h_2 \in H$ and h_0 is not the label of the root of μ *(division rules for for non–elementary membranes).*
- $i_{out} \in H \cup \{env\}$ (if $i_{out} \in H$ then i_{out} is the label of a leaf of μ).

A polarizationless P system with active membranes of degree q,

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

can be viewed as a set of q membranes, labelled by elements of H, arranged in a hierarchical structure μ given by a rooted tree, whose root is called the *skin membrane*, such that: (a) $\mathcal{M}_1, \ldots, \mathcal{M}_q$ represent the finite multisets of *objects* initially placed in the q membranes of the system; (b) $\mathcal{R}$ is a finite set of rules over Γ associated with the labels; (c) $i_{out} \in H \cup \{env\}$ indicates the output region. We use the term *region* i to refer to membrane i in the case $i \in H$ and to refer to the "environment" of the system in the case $i = env$. The leaves of μ are called elementary membranes; any other membrane is said to be non-elementary.

3.2. Polarizationless P systems with active membranes: Semantics

An *instantaneous description* or a *configuration* $\mathcal{C}_t$ at an instant t of a polarizationless P system with active membranes is described by the following elements: (a) the

membrane structure at instant t, and (b) all multisets of objects over Γ associated with all the membranes present in the system at that moment.

An object evolution rule

$$[\, a \to u \,]_h$$

for $h \in H$, $a \in \Gamma$, $u \in M_f(\Gamma)$, is *applicable* to a configuration $\mathcal{C}_t$ at an instant t, if there exists a membrane labelled by h in $\mathcal{C}_t$ which contains object a. When applying such a rule, object a is consumed and objects from multiset u are produced in that membrane.

A send-in communication rule

$$a\,[\ \ ]_h \to [\, b \,]_h$$

for $h \in H$, $a, b \in \Gamma$, is *applicable* to a configuration $\mathcal{C}_t$ at an instant t, if there exists a membrane labelled by h in $\mathcal{C}_t$ such that h is not the label of the root of μ and its parent membrane contains object a. When applying such a rule, object a is consumed from the parent membrane and object b is produced in the corresponding membrane h.

A send-out communication rule

$$[\, a \,]_h \to b\,[\ \ ]_h$$

for $h \in H$, $a, b \in \Gamma$, is *applicable* to a configuration $\mathcal{C}_t$ at an instant t, if there exists a membrane labelled by h in $\mathcal{C}_t$ such that it contains object a. When applying such a rule, object a is consumed from such membrane h and object b is produced in the parent of such membrane.

A dissolution rule

$$[\, a \,]_h \to b$$

for $h \in H$, $a, b \in \Gamma$, is *applicable* to a configuration $\mathcal{C}_t$ at an instant t, if there exists a membrane labelled by h in $\mathcal{C}_t$, different from the skin membrane and from the output region, such that it contains object a. When applying such a rule, object a is consumed, membrane h is dissolved and its objects are sent to the parent (or the first ancestor that has not been dissolved at the same instant t).

A division rule

$$[\, a \,]_h \to [\, b \,]_h [\, c \,]_h$$

is *applicable* to a configuration $\mathcal{C}_t$ at an instant t, if there exists an elementary membrane labelled by h in $\mathcal{C}_t$, different from the skin membrane and from the output region, such that it contains object a. When applying a division rule $[a]_h \to [\, b \,]_h\, [\, c \,]_h$ to a membrane labelled by h in a configuration $\mathcal{C}_t$, under the influence of object a, the membrane with label h 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 are replicated and copies of them are placed in the two new membranes.

A division rule

$$[\,[\,]_{h_1}[\,]_{h_2}\,]_{h_0} \to [\,[\,]_{h_1}\,]_{h_0}\ [\,[\,]_{h_2}\,]_{h_0}$$

is *applicable* to a configuration $\mathcal{C}_t$ at an instant t, if there exists a membrane labelled by h_0 in $\mathcal{C}_t$, different from the skin membrane and from the output region, which contains two membranes labelled by h_1 and h_2, respectively. When applying such a division rule to a membrane labelled by h_0 in a configuration $\mathcal{C}_t$, the membrane with label h_0 is divided into two membranes with the same label containing, respectively, membrane h_1 and membrane h_2 with their contents. Besides, if the membrane labelled by h_0 contains other membranes than those with the labels h_1, h_2, then these membranes are duplicated and are part of the contents of both new copies of the membrane h_0.

In polarizationless P systems with active membranes, the rules are applied according to the following principles:

- The rules associated with membranes labelled with h are used for all copies of this membrane.
- At one transition step, one object of a membrane can be used by only one rule (chosen in a non–deterministic way).
- At one transition step, *a membrane* can be subject of *at most one* rule of types (b_0)–(f_0). Rules of these types can be applied at most once on each membrane at each step.
- Object evolution rules can be simultaneously applied to a membrane joint with one rule of types (b_0)–(f_0). In that case, object evolution rules are applied in a maximally parallel manner.
- If at the same time a membrane labelled with h is divided by a rule of type (e_0) or (f_0) and there are objects in this membrane which evolve by means of rules of type (a_0), then we suppose that first the evolution rules of type (a_0) are used, changing the objects, and then the division is produced. Of course, this process takes only one transition step.
- The skin membrane and the output membrane can never divide or dissolve.

3.3. Polarizationless P systems with active membranes and minimal cooperation in object evolution rules

Next, minimal cooperation in objects evolution rules is introduced in the framework of polarizationless P system with active membranes. The term "minimal cooperation" is used in the following sense: the left-hand side of such rules consists of at most two symbols and its length is greater than or equal to the right-hand side.

Definition 2. A polarizationless P system with active membranes and minimal cooperation in object evolution rules is a polarizationless P system with active membranes such that the object evolution rules are of the following forms:

$$[\,a \to c\,]_h,\ [\,a\,b \to c\,]_h,\ [\,a\,b \to c\,d\,]_h$$

for $h \in H$ and $a, b, c, d \in \Gamma$.

Let us notice that in these systems, send-in communication rules, send-out communication rules, dissolution rules and division rules are non-cooperative rules.

In polarizationless P system with active membranes and minimal cooperation, the rules are applied according to the same principles described in Subsection 3.2.

3.4. Recognizer membrane systems

In this section, a membrane system designates a P system of one of the different variants considered in the paper.

Definition 3. We say that a membrane system Π is a *recognizer* membrane system if the following holds:

(I) The working alphabet Γ of Π has two distinguished objects `yes` and `no`.
(II) There exists an (input) alphabet Σ strictly contained in Γ.
(III) The initial multisets $\mathcal{M}_1, \ldots, \mathcal{M}_q$ of Π are multisets over $\Gamma \setminus \Sigma$.
(IV) There exists a distinguished membrane called the input membrane.
(V) The output region i_{out} is the environment.
(VI) All computations halt.
(VII) 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.

In recognizer membrane systems any computation is either an *accepting computation* (when object `yes` is released into the environment at the last step) or a *rejecting computation* (when object `no` is released into the environment at the last step)

For each finite multiset m over the input alphabet Σ, the *computation of the system* Π *with input* m starts from the configuration obtained by adding the input multiset m to the contents of the input membrane, in the initial configuration of Π. Therefore, in this kind of systems we have an initial configuration associated with each input multiset m (over the input alphabet Σ) . We denote by $\Pi + m$ the membrane system Π with input multiset m.

We also use the following notations:

- $\mathcal{AM}^0(\gamma, \delta)$ where $\gamma \in \{-d, +d\}$ and $\delta \in \{-n, +n\}$, is the class of all recognizer polarizationless P systems with active membranes.
- $\mathcal{AM}^0_{mc}(\gamma, \delta)$ where $\gamma \in \{-d, +d\}$ and $\delta \in \{-n, +n\}$, is the class of all recognizer polarizationless P systems with active membranes with minimal cooperation in object evolution rules.

The meaning of parameters γ and δ is the following:

- if $\gamma = +d$ then dissolution rules are permitted,
- if $\gamma = -d$ then dissolution rules are forbbiden,

- if $\delta = +n$ then division rules for elementary and non–elementary membranes are permitted,
- if $\delta = -n$ then only division rules for elementary membranes are permitted.

3.5. Polynomial complexity classes of recognizer membrane systems

Next, let us recall the concept of efficient solvability by means of a family of recognizer membrane systems (see [12] for more details).

Definition 4. 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, denoted by $X \in \mathbf{PMC}_{\mathcal{R}}$, if the following statements 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 $n \in \mathbb{N}$;
- there exists a pair (cod, s) of polynomial-time computable functions over the set 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.

The polynomial complexity class $\mathbf{PMC}_{\mathcal{R}}$ is closed under polynomial-time reduction and under complement [13].

At the beginning of 2005, Gh. Păun proposed a problem (problem **F** from [10]) which can be formally formulated as follows:

"Is the complexity class $\mathbf{PMC}_{\mathcal{AM}^0(+d,-n)}$ *equal to* $\mathbf{P}$*?"*

That is, in order to provide polynomial time solutions to computationally hard problems by means of families of P systems with active membranes (using only division rules for elementary membranes), can the polarizations be completely avoided? Gh. Păun wrote "*my feeling is that this is not possible*", that is, his guess is that

$$\mathbf{PMC}_{\mathcal{AM}^0(+d,-n)} = \mathbf{P}.$$

By using the dependency graph technique it has been shown that if dissolution rules are forbidden, then only tractable problems can be solved in an efficient way

by families of polarizationless P systems with active membranes even using division rules for elementary and non-elementary membranes, that is,

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

(see [6] for details). In some sense, a partial affirmative answer to Păun's conjecture is obtained in the case that dissolution rules are forbidden.

On the other hand, a family of recognizer polarizationless P systems with active membranes using dissolution rules and division for non–elementary membranes solving `QBF-SAT` in a *uniform* way and in a linear time, has been provided (see [3] for details), and thus

$$\mathbf{PSPACE} \subseteq \mathbf{PMC}_{\mathcal{AM}^0\,(+d,+n)}.$$

In this manner, assuming that $\mathbf{P} \neq \mathbf{NP}$, we have a *partial negative* answer to Păun's conjecture: computationally hard problems can be efficiently solved avoiding polarizations, but the answer is partial in the sense that division rules for non–elementary membranes have been considered.

From the previous result we deduce that when dealing with polarizationless P systems with active membranes including non–elementary membranes division, dissolution rules constitute a key ingredient to solve hard problems efficiently.

4. On efficiency of membrane systems from $\mathcal{AM}^0_{mc}(-d,-n)$

In this section, we show that the syntactical ingredient of minimal cooperation in polarizationless P systems with active membranes (without dissolution and allowing only division for elementary membranes) is enough to solve computationally hard problems in an efficient way.

4.1. A uniform polynomial time solution to `SAT` *problem in* $\mathcal{AM}^0_{mc}(-d,-n)$

Next, a polynomial time solution to `SAT` problem, a well known **NP**-complete problem [5], by a family

$$\mathbf{\Pi} = \{\, \Pi(t) \mid\ t \in \mathbb{N} \,\}$$

of recognizer P systems from $\mathcal{AM}^0_{mc}(-d,-n)$ is provided. Each system $\Pi(t)$ will process any Boolean formula φ in conjunctive normal form with n variables and p clauses, where $t = \langle n,p\rangle$, provided that the appropriate input multiset $cod(\varphi)$ is supplied to the system (through the corresponding input membrane).

Let us recall that the polynomial–time computable function (the *pair function*)

$$\langle n,p\rangle = ((n+p)(n+p+1)/2)+n$$

is a primitive recursive and bijective function from $\mathbb{N}\times\mathbb{N}$ to $\mathbb{N}$. Then, for each $n,p \in \mathbb{N}$, we consider the recognizer P system

$$\Pi(\langle n,p\rangle) = (\Gamma, \Sigma, H, \mu, \mathcal{M}_1, \mathcal{M}_2, \mathcal{R}, i_{in}, i_{out})$$

from $\mathcal{AM}^0_{mc}(-d,-n)$, defined as follows:

(1) Working alphabet:

$$\begin{aligned}\Gamma = \Sigma \cup \{&\texttt{yes}, \texttt{no}, \alpha, \beta', \beta'', \gamma, \gamma', \gamma'', \#\} \\ &\cup \{\, a_{i,k} \mid 1 \le i \le n \wedge 1 \le k \le i \,\} \\ &\cup \{\, \beta_k \mid 0 \le k \le n + 2p \,\} \\ &\cup \{\, t_{i,k}, f_{i,k} \mid 1 \le i \le n-1 \wedge i \le k \le n-1 \,\} \\ &\cup \{\, T_{i,j}, F_{i,j} \mid 1 \le i \le n \wedge 1 \le j \le p \,\} \\ &\cup \{\, c_{j,k} \mid 1 \le j \le p-1 \wedge j \le k \le p-1 \,\} \\ &\cup \{\, c_j \mid 1 \le j \le p \,\} \cup \{\, d_j \mid 2 \le j \le p \,\}\end{aligned}$$

where the input alphabet is $\Sigma = \{\, x_{i,j}, \overline{x}_{i,j}, x^*_{i,j} \mid 1 \le i \le n \wedge 1 \le j \le p \,\}$;

(2) $H = \{1, 2\}$;

(3) membrane structure: $\mu = [\ [\ \ ]_2\ ]_1$, that is, $\mu = (V, E)$ where $V = \{1, 2\}$ and $E = \{(1, 2)\}$;

(4) initial multisets: $\mathcal{M}_1 = \{\, \alpha, \beta_0 \,\}$ and $\mathcal{M}_2 = \{\, a_{1,1}, \ldots a_{n,1} \,\}$

(5) the set of rules $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ consists of the following rules:

Rules in $\mathcal{R}_1$:

1.1

$[\ \alpha\ \gamma\ \longrightarrow\ \gamma'\ ]_1$
$[\ \gamma'\ \longrightarrow\ \gamma''\ ]_1$
$[\ \gamma''\ ]_1\ \longrightarrow\ \texttt{yes}\ [\ \ ]_1$

1.2

$[\ \beta_k\ \longrightarrow\ \beta_{k+1}\ ]_1$, for $0 \le k \le n + 2p - 1$
$[\ \beta_{n+2p}\ \longrightarrow\ \beta'\ ]_1$
$[\ \alpha\ \beta'\ \longrightarrow\ \beta''\ ]_1$
$[\ \beta''\ ]_1\ \longrightarrow\ \texttt{no}\ [\ \ ]_1$

Rules in $\mathcal{R}_2$:

2.1

$[\ a_{i,i}\ ]_2 \longrightarrow [\ t_{i,i}\ ]_2\ [\ f_{i,i}\ ]_2$, for $1 \le i \le n-1$
$[a_{i,k} \longrightarrow\ a_{i,k+1}\ ]_2$, for $2 \le i \le n \wedge 1 \le k \le i-1$
$[\ a_{n,n}\ ]_2 \longrightarrow [\ T_{n,1}\ ]_2\ [\ F_{n,1}\ ]_2$

2.2

$$\left.\begin{array}{l}[t_{i,k} \longrightarrow\ t_{i,k+1}\ ]_2\\ [f_{i,k} \longrightarrow\ f_{i,k+1}\ ]_2\end{array}\right\} 1 \leq i \leq n-2 \wedge i \leq k \leq n-2$$

$$\left.\begin{array}{l}[t_{i,n-1} \longrightarrow\ T_{i,1}\ ]_2\\ [f_{i,n-1} \longrightarrow\ F_{i,1}\ ]_2\end{array}\right\} 1 \leq i \leq n-1$$

2.3

$$\left.\begin{array}{l}[T_{i,j}\ x_{i,j}\ \longrightarrow\ T_{i,j+1}\ c_{j,j}\ ]_2\\ [T_{i,j}\ \overline{x}_{i,j}\ \longrightarrow\ T_{i,j+1}\ ]_2\\ [T_{i,j}\ x^*_{i,j}\ \longrightarrow\ T_{i,j+1}\ ]_2\\ [F_{i,j}\ x_{i,j}\ \longrightarrow\ F_{i,j+1}\ ]_2\\ [F_{i,j}\ \overline{x}_{i,j}\ \longrightarrow\ F_{i,j+1}\ c_{j,j}\ ]_2\\ [F_{i,j}\ x^*_{i,j}\ \longrightarrow\ F_{i,j+1}\ ]_2\end{array}\right\} 1 \leq i \leq n \wedge 1 \leq j \leq p-1$$

$$\left.\begin{array}{l}[T_{i,p}\ x_{i,p}\ \longrightarrow\ c_p\ ]_2\\ [T_{i,p}\ \overline{x}_{i,p}\ \longrightarrow\ \#\ ]_2\\ [T_{i,p}\ x^*_{i,p}\ \longrightarrow\ \#\ ]_2\\ [F_{i,p}\ x_{i,p}\ \longrightarrow\ \#\ ]_2\\ [F_{i,p}\ \overline{x}_{i,p}\ \longrightarrow\ c_p\ ]_2\\ [F_{i,p}\ x^*_{i,p}\ \longrightarrow\ \#\ ]_2\end{array}\right\} 1 \leq i \leq n$$

2.4

$$[c_{j,k}\ \longrightarrow\ c_{j,k+1}\ ]_2, \text{ for } 1 \leq j \leq p-2,\ j \leq k \leq p-2$$
$$[c_{j,p-1}\ \longrightarrow\ c_j\ ]_2, \text{ for } 1 \leq j \leq p-1$$

2.5

$$[c_1\ c_2\ \longrightarrow\ d_2\ ]_2$$
$$[d_j\ c_{j+1}\ \longrightarrow\ d_{j+1}\ ]_2\,, \text{ for } 2 \leq j \leq p-1$$

2.6

$$[\ d_p\ ]_2\ \longrightarrow\ \gamma\ [\ \ ]_2$$

(6) the input membrane is the membrane labelled by 2 ($i_{in} = 2$) and the output region is the environment ($i_{out} = env$).

4.2. An overview of the computations

Let $\varphi = C_1 \wedge \cdots \wedge C_p$ be an instance of the `SAT` problem consisting of p clauses

$$C_j = l_{j,1} \vee \cdots \vee l_{j,r_j},\ 1 \leq j \leq p,$$

where $Var(\varphi) = \{x_1, \cdots, x_n\}$, and

$$l_{j,k} \in \{\, x_i, \neg x_i \mid 1 \leq i \leq n \,\},\ 1 \leq j \leq p, 1 \leq k \leq r_j.$$

Let us assume that the number of variables, n, and the number of clauses, p, of φ, are greater than or equal to 2.

We consider the polynomial encoding (cod, s) from SAT in $\mathbf{\Pi}$ defined as follows: for each Boolean formula φ in conjunctive normal form with n variables and p clauses, $s(\varphi) = \langle n, p\rangle$ and

$$cod(\varphi) = \{\, x_{i,j} \mid x_i \in C_j \,\} \cup \{\, \overline{x}_{i,j} \mid \neg x_i \in C_j \,\} \cup \{\, x^*_{i,j} \mid x_i \notin C_j \wedge \neg x_i \notin C_j \,\}.$$

The Boolean formula φ will be processed by the system $\Pi(s(\varphi)) + cod(\varphi)$.

For instance, the formula $\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_4) \wedge (x_1 \vee \neg x_2 \vee \neg x_4)$ is encoded as follows[1]:

$$cod(\varphi) = \begin{pmatrix} x_{1,1} & x_{2,1} & \overline{x}_{3,1} & x^*_{4,1} \\ x^*_{1,2} & \overline{x}_{2,2} & x^*_{3,2} & x_{4,2} \\ x_{1,3} & \overline{x}_{2,3} & x^*_{3,3} & \overline{x}_{4,3} \end{pmatrix}$$

That is, each four elements represent a clause of φ.

The proposed solution follows a brute force algorithm implemented in the framework of recognizer polarizationless P systems with active membranes and minimal cooperation. The solution consists of the following stages:

- *Generation stage*: using division rules, all truth assignments for the variables $\{x_1, \ldots, x_n\}$ associated with φ are produced. Specifically, 2^n membranes labelled by 2 are generated, each of them encoding a truth assignment. This stage takes exactly n computation steps, where n is the number of variables of φ.
- *First Checking stage*: checking whether or not each clause of the input formula φ is satisfied by the truth assignment generated in the previous stage, encoded by a membrane labelled by 2. This stage takes exactly p steps, where p is the number of clauses of φ.
- *Second Checking stage*: checking whether or not all clauses of the input formula φ are satisfied by some truth assignment encoded by a membrane labelled by 2. This stage takes exactly $p - 1$ steps.
- *Output stage*: the system sends to the environment the right answer according to the results of the previous stage. The output stage takes 4 steps.

4.3. Generation stage

Here, the goal is to generate all truth assignments for the variables associated with the Boolean formula $\varphi(x_1, \ldots, x_n)$, by applying division rules from **2.1** in membranes labelled by 2. In the i–th step $(1 \leq i \leq n-1)$ of this stage, division rule associated with object $a_{i,i}$ is triggered and objects $t_{i,i}, f_{i,i}$ are produced in the new created

[1] We display the multiset $cod(\varphi)$ in matrix form to highlight the correspondence between objects and clauses.

membranes labelled by 2. In the last step of this stage, the objects produced are $T_{n,1}$ and $F_{n,1}$.

We use the rules

$$[a_{i,k} \to a_{i,k+1}]_2,\ 2 \leq i \leq n \wedge 1 \leq k \leq i-1$$

in order to produce the next $a_{j,j}$, where $j = k+1$, and in the following step, we can apply the j-th division rule. We do it $n-1$ times. Beside this, we use rules from **2.2** to have, in the $(n-1)$-th step, objects $t_{i,n-1}, f_{i,n-1}, 1 \leq i \leq n-1$, depending on the truth assignment associated with each membrane. At the end of this stage, the objects produced in membranes labelled by 2 in configuration $\mathcal{C}_n$ are $T_{i,1}$ and $F_{i,1}$, with $1 \leq i \leq n$, respectively.

4.4. First Checking stage

At this stage, we try to determine which clauses are satisfied for the truth assignment encoded by each membrane labelled by 2. For that, rules from **2.3** will be applied in such manner that in the j-th step $(1 \leq j \leq p)$ of this stage, clause j is checked and an object c_j is produced only if the clause C_j is satisfied.

If there is an object $T_{i,j}$ (resp., $F_{i,j}$), it means that in the truth assignment of this membrane the value of x_i is true (resp., false), and that the clause j is being checked. If there is an object $x_{i,j}$ (resp., $\overline{x}_{i,j}$), then clause j is satisfied by variable i, and an object c_j is generated to witness it. At the end of this stage, at configuration $\mathcal{C}_{n+p}$, an object c_j is in a membrane labelled by 2 if and only if the truth assignment encoded by that membrane makes true clause C_j.

4.5. Second Checking stage

At this stage, we try to determine if some truth assignment encoded by a membrane labelled by 2 satified all clauses of the input formula. For that, rules from **2.4** will be applied in such manner that object d_j $(2 \leq j \leq p)$ is produced in the case clauses $c_1, \ldots, c_j$ are satisfied. Therefore, at the end of this stage, at configuration $\mathcal{C}_{n+2p-1}$, an object d_p appears in a membrane labelled by 2 if and only if the truth assignment encoded by that membrane makes true the input formula φ.

4.6. Output stage

The output stage takes 4 steps and the answer of the system is encoded in the environment associated with configuration $\mathcal{C}_{n+2p+3}$.

- *Affirmative answer*: if the truth assignment encoded by any membrane with label 2 makes true all clauses, then an object d_p will appear in that membrane at configuration $\mathcal{C}_{n+2p-1}$. In this case, by applying rule from **2.5** an object γ is produced in the skin membrane and, simultaneously, by applying rules in **1.2**, we have $\beta_{n+2p} \in \mathcal{C}_{n+2p}(1)$, where $\mathcal{C}_j(i)$ denotes the multiset contained in membrane i at configuration $\mathcal{C}_j$.

At step $n + 2p + 1$, objects γ' and β' are simultaneously produced in the skin membrane. At the next step, object γ' evolves into object γ'' in the skin membrane at configuration $\mathcal{C}_{n+2p+2}$. Finally, object γ'' sends out object `yes` to the environment, resulting in the configuration $\mathcal{C}_{n+2p+3}$, and the computation halts.

– *Negative answer*: if none of the truth assignments encoded by membranes with label 2 makes the formula φ true, then object d_p does not appear in any membrane with label 2. Thus, only a rule from **1.2** is applicable to configuration $\mathcal{C}_{n+2p-1}$ and we have $\mathcal{C}_{n+2p}(1) = \{\alpha\,,\,\beta_{n+2p}\}$. At the next step, by applying rule $[\,\beta_{n+2p} \longrightarrow \beta'\,]_1$ we have $\mathcal{C}_{n+2p+1}(1) = \{\,\alpha\,,\,\beta'\,\}$. Then, by applying rule $[\,\alpha\,\beta' \longrightarrow \beta''\,]_1$ an object β'' is produced in the skin membrane at configuration $\mathcal{C}_{n+2p+2}$. Finally, object β'' sends out object `no` to the environment, resulting in the configuration $\mathcal{C}_{n+2p+3}$, and the computation halts.

4.7. Polynomial Uniformity of the Family

In this subsection, we show that the family $\mathbf{\Pi} = \{\,\Pi(t) \mid t \in \mathbb{N}\,\}$ defined above is polynomially uniform by Turing machines. To this aim we prove that $\Pi(\langle n, p\rangle)$ is built in polynomial time with respect to the size parameters n and p of `SAT` problem instances.

It is easy to check that the rules of a system $\Pi(\langle n, p\rangle)$ of the family are recursively defined from the values n and p. The amount of resources to build an element of the family is of a polynomial order in the number of variables n and the number of clauses p, as shown below:

(I) Size of the alphabet:

$$5np + \frac{3n^2 + p^2 + n + 7p}{2} + 9 \in \Theta((max\{n, p\})^2).$$

(II) Initial number of cells:

$$2 \in \Theta(1).$$

(III) Initial number of objects in cells:

$$n + 2 \in \Theta(n).$$

(IV) Number of rules:

$$6np + \frac{3n^2 + p^2 + n + 5p}{2} + 6 \in \Theta((max\{n, p\})^2).$$

(V) Maximal number of objects involved in any rule:

$$4 \in \Theta(1).$$

Let us recall that the total number of literals in a formula in conjunctive normal form is of the order $O(n \cdot p)$. Therefore, there exists a deterministic Turing machine that builds the system $\Pi(\langle n, p\rangle)$ in a polynomial time with respect to n and p.

From this, we conclude:

Theorem 5. $\texttt{SAT} \in \mathbf{PMC}_{\mathcal{AM}^0_{mc}(-d,-n)}$.

Consequently, and having in mind that the complexity class $\mathbf{PMC}_{\mathcal{AM}^0_{mc}(-d,-n)}$ is closed under polynomial-time reduction and complement, we have the following result.

Corollary 6. $\mathbf{NP} \cup \mathbf{co} - \mathbf{NP} \subseteq \mathbf{PMC}_{\mathcal{AM}^0_{mc}(-d,-n)}$.

5. Conclusions

The classical definition of polarizationless P systems with active membranes makes use of non-cooperative rules and their object evolution rules are of the form $[\, a \rightarrow u \,]_h$, where a is an object and u is a finite multiset of objects. It is well known [6] that only tractable problems can be solved in an efficient way by families of such kind of P systems when division for elementary and non-elementary membranes are permitted but dissolution rules are forbidden, that is,

$$\mathbf{P} = \mathbf{PMC}_{\mathcal{AM}^0(-d,+n)}.$$

In this paper, polarizationless P systems with active membranes and minimal cooperation in object evolution rules are introduced. This kind of rules are of the forms

$$[a \rightarrow c]_h,\ [ab \rightarrow c]_h \text{ or } [ab \rightarrow cd]_h.$$

The computational efficiency of this model is studied, and a uniform polynomial-time solution to SAT problem by a family of polarizationless P systems with active membranes, without dissolution rules and minimal cooperation in object evolution rules, using only division for elementary membranes, is provided. Consequently, in the framework of polarizationless P systems with active membranes, without dissolution rules and using only division for elementary membranes, a frontier between efficiency and non-efficiency is obtained when passing from non-cooperation to minimal cooperation in object evolution rules.

As future work, we propose to study the capability of the new framework to provide uniform polynomial-time solution to **PSPACE**-complete problems as well as to analyse these membrane systems when separation rules are considered instead of division rules, that is, when distribution of objects between the two created membranes (according to a partition of the working alphabet) is considered instead of replication of objects.

Acknowledgement

Luis Valencia-Cabrera acknowledges the support of the projects Grant No. 61033003 and 61320106005 of the National Natural Science Foundation of China.

References

[1] A. Alhazov, L. Pan, Polarizationless P systems with active membranes. *Grammars* **7** (2004), 141–159.

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

[3] A. Alhazov, M. J. Pérez-Jiménez, Uniform solution of QSAT using polarizationless active membranes. In: J. Durand-Lose, M. Margenstern (eds.), *Machines, Computations, and Universality*. LNCS 4664, Springer-Verlag, 2007, 122–133.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, *An Introduction to Algorithms*. The MIT Press, Cambridge, Massachussets, 1994.

[5] M. R. Garey, D. S. Johnson, *Computers and Intractability. A guide to the theory of NP-completeness*. W. H. Freeman and Company, New York, 1979.

[6] M. A. Gutiérrez-Naranjo, M. J. Pérez-Jiménez, A. Riscos-Núñez, F. J. Romero-Campero, On the power of dissolution in P systems with active membranes. In: R. Freund, Gh. Păun, G. Rozenberg, A. Salomaa (eds.), *Membrane Computing, 6th International Workshop, WMC 2005, Vienna, Austria, July 18–21, 2005, Revised Selected and Invited Papers*. LNCS 3850, Springer-Verlag, 2006, 224–240.

[7] M. A. Gutiérrez-Naranjo, M. J. Pérez-Jiménez, A. Riscos-Núñez, F. J. Romero-Campero, A. Romero-Jiménez, 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, 137–154.

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

[9] Gh. Păun, P systems with active membranes: Attacking **NP**-complete problems. *Journal of Automata, Languages and Combinatorics* **6** (2001) 1, 75–90.

[10] Gh. Păun, Further twenty six open problems in membrane computing. In: M. A. Gutiérrez, A. Riscos, F. J. Romero, D. Sburlan (eds.), *Proceedings of the Third Brainstorming Week on Membrane Computing*. Report RGNC 01/04, Fénix Editora, Sevilla, 2005, 249–262.

[11] Gh. Păun, G. Rozenberg, A. Salomaa (eds.), *The Oxford Handbook of Membrane Computing*. Oxford University Press, Oxford, 2010.

[12] M. J. Pérez-Jiménez, An approach to computational complexity in Membrane Computing. In: G. Mauri, Gh. Păun, M. J. Pérez-Jiménez, G. Rozenberg, A. Salomaa (eds.), *Membrane Computing, International Workshop, WMC5, Milano, Italy, 2004, Selected Papers*. LNCS 3365, Springer-Verlag, 2005, 85–109.

[13] M. J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini, A polynomial complexity class in P systems using membrane division. *Journal of Automata, Languages and Combinatorics* **11** (2006) 4, 423–434.

[14] A. E. Porreca, G. Mauri, C. Zandron, Complexity classes for membrane systems. *Informatique théorique et applications*, **40** (2006) 2, 141–162.

[15] P. Sosík, A. Rodríguez-Patón, P systems with active membranes characterize PSPACE. In: Ch. Mao, T. Yokomori (eds.), *DNA Computing, 12th International Meeting on DNA Computing, DNA12, Seoul, 2006, Revised Selected Papers*. LNCS 4287, Springer-Verlag, 2006, 33–46.

(Received: June 2, 2016; revised: August 8, 2016)