Reaching efficiency through collaboration in membrane systems: Dissolution, polarization and cooperation

Luis Valencia-Cabrera, David Orellana-Martín, Miguel A. Martínez-del-Amor, Agustín Riscos-Núñez, Mario J. Pérez-Jiménez

Research Group of Natural Computing, Department of Computer Science and Artificial Intelligence, Universidad de Sevilla, Sevilla, 41012, Spain

Keywords:
Membrane computing
Active membranes
Cooperative rules Minimal cooperation
Computational complexity
The **P** versus **NP** problem

A B S T R A C T

From a computational complexity point of view, some syntactical ingredients play different roles depending on the kind of combination considered. Inspired by the fact that the passing of a chemical substance through a biological membrane is often done by an interaction with the membrane itself, systems with active membranes were considered. Several combinations of different ingredients have been used in order to know which kind of problems could they solve *efficiently* In this paper, minimal cooperation with a minimal expression (the left-hand side of every object evolution rule has at most two objects and its right-hand side contains only one object) in object evolution rules is considered and a polynomial-time uniform solution to the SAT problem is presented. Consequently, a new way to tackle the **P** versus **NP** problem is provided.

1. Introduction

The first models in Membrane Computing were designed in such a manner that the number of membranes could not increase during a computation. They could only decrease by dissolving membranes as a result of applying some rules to the objects present in the system. However, in these systems an exponential workspace (expressed in terms of the number of objects) can be constructed in linear time, e.g. via evolution rules of the type $a \to (a^2, here)$. Nevertheless, such a capability is not enough to efficiently solve **NP**-complete problems, unless $\mathbf{P} = \mathbf{NP}$ (see [6] for details).

It is well known that in ideal circumstances, a cell produces two identical copies by the processes of *interphase* and *mitosis*. First, the cell grows and makes a copy of its DNA (replication) and, finally, the cell separates its DNA into two sets and divides its cytoplasm, forming two new cells. Inspired from this mechanism, a new kind of rules was introduced in Membrane Computing allowing the proliferation of membranes by means of division rules. A membrane without any other membrane inside it (*elementary* membrane) can be divided by means of an interaction with an object from that membrane. In [12], cell-like P systems with active membranes are defined incorporating this ingredient. In such systems, each membrane has an electrical charge (positive, negative or neutral) associated with it, but the rules are non-cooperative and there are no priorities among rules. Besides, a non-elementary membrane (that is, a membrane with one or more membranes within it) with at least two inner membranes can also be divided. The skin cannot be divided.

In the classical version of these systems, each membrane is identified by a label and has an electrical polarization (positive, negative or neutral) associated with it. They can be dissolved or their objects can be replicated (respectively distributed) by a mechanism abstracting the cell division (membrane fission) process. In each membrane, objects can evolve according to given evolution rules, or can be sent outside the current membrane or to an inner membrane. This framework is too powerful with respect to efficiency, so that **PSPACE**-complete problems can be solved in polynomial time with respect to the input size by avoiding cooperative rules, dissolution and division for non-elementary membranes.

The class of all problems solvable in polynomial time with respect to the input size and in a uniform way[1] by means of families of P systems with active membranes using division for elementary and non-elementary membranes contains class **PSPACE** and it is contained in class **EXP** [15]. Thus, in order to provide efficient solutions to computationally hard problems, this framework seems to be too powerful from the computational complexity point of view. If polarizations are removed, then dissolution rules become essential, in the sense that, without their presence, only problems in class **P** can be efficiently solved by families of these kind of systems.

P systems with active membranes and without electrical charges were initially studied in [1,2]. Polarizations were replaced by the possibility to change the label of the membranes by means of some rules. However, in order to obtain polynomial-time solutions to computationally hard problems, two polarizations suffice (see [3] for details). In [14] bi-stable catalysts are used to compensate the loss of computational efficiency represented by avoiding polarizations. Recognizer polarizationless P systems with active membranes were introduced in [7] as systems such that: (a) the working alphabet has two distinguished objects $\mathtt{yes}$ and $\mathtt{no}$; (b) there is an input alphabet strictly contained in the working alphabet; (c) the initial multisets do not contain any object from the input alphabet; (d) there is a distinguished membrane (the input membrane); (e) the output region is the environment; (f) all computations halt; and (g) either object $\mathtt{yes}$ or object $\mathtt{no}$ (but not both) must have been released into the environment for any computation, and only at the last step.

In [7] the first polynomial-time semi-uniform solution to $\mathtt{SUBSET\text{-}SUM}$ problem was given in the framework of polarizationless P systems with active membranes using division rules for elementary and non-elementary membranes and without dissolution. In [4] a polynomial-time uniform solution to the satisfiability of a quantified Boolean formula ($\mathtt{QSAT}$) problem (a well known **PSPACE**-complete problem) was given in that framework by using division for elementary and non-elementary membranes. In [7] was highlighted the surprising role of (up to then) an apparently "innocent" type of rules: dissolution. Thus, in the framework of polarizationless P systems with active membranes using division for elementary and non-elementary membranes, passing from allowing dissolution rules to forbidding them amounts to passing from efficiency to non-efficiency.

Let us recall that in P systems with active membranes, division rules provide a mechanism to produce an exponential number of membranes in linear time. This mechanism can be replaced by other one, inspired by the membrane fission process, by which a biological membrane splits into two new ones in such a manner that the contents of the initial membrane is *distributed* between the new membranes. Membrane separation rules have been introduced in the framework of P systems with active membranes instead of division rules, and their efficiency have been shown [8–10].

Rules involving the evolution of several objects at the same time are called *cooperative rules*. It is the case of cooperation by means of objects. An object a needs additional objects in order to evolve, and the application of the rules "consumes" the additional objects used. An interesting particular case of cooperative rules is the ones involving certain special objects (called *catalysts*), specified in advance. In these rules, regular objects only evolve in evolution rules in conjunction with these catalyst objects, which remain unmodified [11].

It is worth noting that polarization in P systems with active membranes also provides a certain kind of "cooperation". In these systems, a rule associated with the label of a membrane can be triggered at a moment of time t when a certain object is present in the membrane. However, additionally, it needs to "cooperate" with a specific polarization associated with the membrane at instant t. In this case, the polarization is not "consumed" by the application of the rule, in the sense that many different objects of the current contents can evolve at instant t "by using" the same polarization.

The paper is organized as follows. In Section 2, polarizationless P systems with active membranes by using minimal cooperation in object evolution rules are considered. The computational efficiency of such P systems using minimal cooperation with minimal production (only a single object is produced by the application of any rule) is established in Section 3 by providing a uniform polynomial-time solution to the $\mathtt{SAT}$ problem. A formal verification of this result is presented in Section 4. Next, the main results of the paper are presented in Section 5. Finally, the paper ends with some conclusions and open problems.

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

We assume the reader is familiar with basic notions and terminology of membrane computing [13]. However, before going on let us briefly overview some notations that will be used throughout the paper. An *alphabet* Γ is a non-empty set whose elements are called *symbols*. 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 $supp(m) = \{x \in \Gamma \mid f(x) > 0\}$, and its

[1] For more details about the uniformity of a solution, see [4].

size is $|m| = \Sigma_{x \in supp(m)} f(x)$. We denote by $M(\Gamma)$ the set of all multisets over Γ. If $m_1 = (\Gamma, f_1)$, $m_2 = (\Gamma, f_2)$ 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 $x \in \Gamma$.

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

Definition 2.1. In the context of polarizationless P system with active membranes, several types of minimal cooperation in object evolution rules are defined as follows.

– *Minimal cooperation* (**mc**): object evolution rules are of the form $[\, u \to v \,]_h$, where $u, v \in M(\Gamma)$ such that $|u| \leq 2$, but at least one object evolution rule verifies $|u| = 2$.
– *Primary minimal cooperation* (**pmc**): object evolution rules are of the form $[\, u \to v \,]_h$, where $u, v \in M(\Gamma)$ and $1 \leq |u|, |v| \leq 2$, but at least one object evolution rule verifies $|u| = 2$.
– *Bounded minimal cooperation* (**bmc**): object evolution rules are of the form $[\, u \to v \,]_h$, where $u, v \in M(\Gamma)$ and $1 \leq |v| \leq |u| \leq 2$, but at least one object evolution rule verifies $|u| = 2$.
– *Minimal cooperation and minimal production* (**mcmp**): object evolution rules are of the forms $[\, a \to b \,]_h$ or $[\, a\, b \to c \,]_h$, where $a, b, c \in \Gamma$, but at least one object evolution rule is of the second type.

In these systems, send-in communication rules, send-out communication rules, dissolution rules and division rules are non-cooperative rules.

In polarizationless P systems with active membranes and minimal cooperation in object evolution rules, the rules are applied according to the same principles than in the "classical" P systems with active membranes (see [12], for details).

We denote by $\mathcal{DAM}^0(\alpha, \beta, \gamma, \delta)$ the class of all recognizer polarizationless P systems with active membranes and division rules, where α, β, γ and δ are parameters associated with object evolution rules, communication rules, dissolution rules and division rules, respectively. The meaning of the parameters $\alpha, \beta, \gamma, \delta$ is the following:

- If $\alpha = -e$ (resp. $\alpha = +e$), object evolution rules are forbidden (resp. permitted).
- If $\alpha = mc$ (resp. $\alpha = pmc$, $\alpha = bmc$ or $\alpha = mcmp$), then minimal cooperation (primary minimal cooperation, bounded minimal cooperation or minimal cooperation and minimal production, respectively) in object evolution rules are permitted.
- If $\beta = -c$ (resp. $\beta = +c$) then communication rules are forbidden (resp. permitted).
- If $\gamma = -d$ (resp. $\gamma = +d$) then dissolution rules are forbidden (resp. permitted).
- If $\delta = -n$ (resp. $\delta = +n$) then division rules for only elementary membranes are permitted (resp. division rules for elementary and non-elementary membranes are permitted).

If separation rules are considered instead of division rules, the corresponding classes of recognizer membrane systems are denoted by $\mathcal{SAM}^0(\alpha, \beta, \gamma, \delta)$.

Let us recall some interesting results expressed in these notations.

1. Families of systems from $\mathcal{DAM}^0(+e, +c, +d, +n)$ can solve **PSPACE**-complete problems in polynomial time and in a uniform way, that is, **PSPACE** $\subseteq$ $\mathbf{PMC}_{\mathcal{DAM}^0(+e,+c,+d,+n)}$ (see [4] for details). Moreover, in [17] and [16] the reverse inclusion was proved, so a stronger result is obtained in this framework: **PSPACE** $=$ $\mathbf{PMC}_{\mathcal{DAM}^0(+e,+c,+d,+n)}$.
2. Families of systems from $\mathcal{DAM}^0(+e, +c, -d, +n)$ can efficiently solve only problems in class **P**, that is, $\mathbf{PMC}_{\mathcal{DAM}^0(+e,+c,-d,+n)} = \mathbf{P}$ (see [7] for details).
3. Families of systems from $\mathcal{SAM}^0(+e, +c, -d, +n)$ can efficiently solve only problems in class **P**, that is, $\mathbf{PMC}_{\mathcal{SAM}^0(+e,+c,-d,+n)} = \mathbf{P}$ (see [19] for details).
4. Families of systems from $\mathcal{DAM}^0(bmc, +c, -d, -n)$ can solve **NP**-complete problems in polynomial time and in a uniform way, that is, **NP** $\cup$ **co-NP** $\subseteq$ $\mathbf{PMC}_{\mathcal{DAM}^0(bmc,+c,-d,-n)}$ (see [18] for details).
5. Families of systems from $\mathcal{SAM}^0(bmc, +c, -d, -n)$ can efficiently solve only problems in class **P**, that is, $\mathbf{PMC}_{\mathcal{SAM}^0(bmc,+c,+d,+n)} = \mathbf{P}$ (see [20] for details).
6. Families of systems from $\mathcal{SAM}^0(pmc, +c, -d, -n)$ can solve **NP**-complete problems in polynomial time and in a uniform way, that is, **NP** $\cup$ **co-NP** $\subseteq$ $\mathbf{PMC}_{\mathcal{SAM}^0(pmc,+c,-d,-n)}$ (see [18] for details).

Păun's conjecture can be expressed as follows: $\mathbf{PMC}_{\mathcal{DAM}^0(+e,+c,+d,-n)} = \mathbf{P}$. It is a relevant open question.

From results 4 and 5, a new frontier of efficiency is deduced. In the framework of polarizationless P systems with active membranes not using dissolution rules, a new frontier of the efficiency is obtained. Specifically, when bounded minimal cooperation in object evolution rules is allowed in the previous computing framework, passing from separation rules to division rules amounts to passing from non-efficiency to efficiency.

Next, we try to find narrower frontiers of efficiency by showing that bounded minimal cooperation can be replaced by minimal cooperation and minimal production in object evolution rules. Let us recall that the application of this kind of rules implies that only a single object can be produced by the application of a rule.

3. On the efficiency of systems from $\mathcal{DAM}^0(mcmp, +c, -d, -n)$

In this section, we analyze what happens, from a computational complexity point of view, when minimal cooperation and minimal production in object evolution rules are considered instead of dissolution rules. The first work in this direction was addressed in [18]. The efficiency of such systems was proven when only object evolution rules such that the of length their left-hand sides are greater than or equal to the length of the corresponding right-hand side, and both lengths are at most two. Specifically, we show that the syntactical ingredient of minimal cooperation and minimal production 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.

3.1. A uniform polynomial-time solution to the SAT problem

Next, a polynomial-time uniform solution to the SAT problem, a well known **NP**-complete problem [5] is provided by a family $\mathbf{\Pi} = \{\Pi(t) \mid t \in \mathbb{N}\}$ of recognizer P systems from $\mathcal{DAM}^0(mcmp, +c, -d, -n)$.

Let us recall that the polynomial-time computable function (the *Cantor 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 $\mathbb{N}$. The family $\mathbf{\Pi} = \{\Pi(t) \mid t \in \mathbb{N}\}$ is defined in such a manner that system $\Pi(t)$ will process any Boolean formula φ in conjunctive normal form (CNF) 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), and will answer if there exists at least one truth assignment that makes true the input formula φ, that is, $\Pi(t)$ will solve an instance of the SAT problem.

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 $\mathcal{DAM}^0(mcmp, +c, -d, -n)$, defined as follows:

(1) Working alphabet Γ:
$\{\mathtt{yes}, \mathtt{no}, \alpha, \beta', \beta'', \gamma, \gamma', \gamma'', \#\} \cup \{a_{i,k} \mid 1 \le i \le n, 1 \le k \le i\} \cup$
$\{t_{i,k}, f_{i,k} \mid 1 \le i \le n, i \le k \le n+p-1\} \cup$
$\{\beta_k \mid 0 \le k \le n+2p+1\} \cup \{c_j \mid 1 \le j \le p\} \cup \{d_j \mid 2 \le j \le p\} \cup$
$\{T_{i,k}, F_{i,k} \mid 1 \le i \le n, 0 \le k \le n-1\} \cup \{T_i, F_i \mid 1 \le i \le n\} \cup$
$\{x_{i,j,k}, \overline{x}_{i,j,k}, x^*_{i,j,k} \mid 0 \le i \le n, 1 \le j \le p, 1 \le k \le n+p\}$
(2) Input alphabet Σ: $\{x_{i,j,0}, \overline{x}_{i,j,0}, x^*_{i,j,0} \mid 1 \le i \le n, 1 \le j \le p\}$.
(3) $H = \{1, 2\}$.
(4) Membrane structure: $\mu = [\,[\;]_2\,]_1$, that is, $\mu = (V, E)$ where $V = \{1, 2\}$ and $E = \{(1, 2)\}$.
(5) Initial multisets: $\mathcal{M}_1 = \{\alpha, \beta_0\}$, $\mathcal{M}_2 = \{a_{i,1}, T^p_{i,0}, F^p_{i,0} \mid 1 \le i \le n\}$.
(6) The set of rules $\mathcal{R}$ consists of the following rules:

1.1 Rules for a general counter.
$[\,\beta_k \longrightarrow \beta_{k+1}\,]_1$, for $0 \le k \le n+2p$
$[\,\beta_{n+2p+1} \longrightarrow \beta'\,]_1$

1.2 Rules for an affirmative answer.
$[\,\alpha\,\gamma \longrightarrow \gamma'\,]_1$
$[\,\gamma' \longrightarrow \gamma''\,]_1$
$[\,\gamma''\,]_1 \longrightarrow \mathtt{yes}\,[\;]_1$

1.3 Rules for a negative answer.
$[\,\alpha\,\beta' \longrightarrow \beta''\,]_1$
$[\,\beta''\,]_1 \longrightarrow \mathtt{no}\,[\;]_1$

2.1 Rules to generate all truth assignments.
$[\,a_{i,i}\,]_2 \longrightarrow [\,t_{i,i}\,]_2\,[\,f_{i,i}\,]_2$, for $1 \le i \le n$
$[a_{i,k} \longrightarrow a_{i,k+1}\,]_2$, for $2 \le i \le n \wedge 1 \le k \le i-1$

2.2 Rules to produce exactly p copies of each truth assignment.
$$\left.\begin{array}{l}[t_{i,k} \longrightarrow t_{i,k+1}\,]_2\\ [f_{i,k} \longrightarrow f_{i,k+1}\,]_2\end{array}\right\} 1 \le i \le n-1 \wedge i \le k \le n-1$$
$$\left.\begin{array}{l}[T_{i,k} \longrightarrow T_{i,k+1}\,]_2\\ [F_{i,k} \longrightarrow F_{i,k+1}\,]_2\end{array}\right\} 1 \le i \le n, 0 \le k \le n-2$$
$$\left.\begin{array}{l}[T_{i,n-1} \longrightarrow T_i\,]_2\\ [F_{i,n-1} \longrightarrow F_i\,]_2\end{array}\right\} 1 \le i \le n$$

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

$$\left.\begin{array}{l}[t_{i,n+p-1}\ F_i \longrightarrow\ \#\ ]_2\\ [f_{i,n+p-1}\ T_i \longrightarrow\ \#\ ]_2\end{array}\right\} 1 \le i \le n$$

2.3 Rules to prepare the input formula for check clauses:

$$\left.\begin{array}{l}[\ x_{i,j,k} \longrightarrow x_{i,j,k+1}\ ]_2\\ [\ \overline{x}_{i,j,k} \longrightarrow \overline{x}_{i,j,k+1}\ ]_2\\ [\ x^*_{i,j,k} \longrightarrow x^*_{i,j,k+1}\ ]_2\end{array}\right\} 1 \le i \le n,\ \ 1 \le j \le p,\ \ 0 \le k \le n+p-1$$

2.4 Rules for the first checking stage.

$$\left.\begin{array}{l}[T_i\ x_{i,j,n+p}\ \longrightarrow\ c_j\ ]_2\\ [T_i\ \overline{x}_{i,j,n+p}\ \longrightarrow\ \#\ ]_2\\ [T_i\ x^*_{i,j,n+p}\ \longrightarrow\ \#\ ]_2\\ [F_i\ x_{i,j,n+p}\ \longrightarrow\ \#\ ]_2\\ [F_i\ \overline{x}_{i,j,n+p}\ \longrightarrow\ c_j\ ]_2\\ [F_i\ x^*_{i,j,n+p}\ \longrightarrow\ \#\ ]_2\end{array}\right\} 1 \le i \le n \wedge 1 \le j \le p$$

2.5 Rules for the second checking stage.

$[c_1\ c_2\ \longrightarrow\ d_2\ ]_2$

$[d_j\ c_{j+1}\ \longrightarrow\ d_{j+1}\ ]_2$, for $2 \le j \le p-1$

2.6 Rule to prepare an affirmative answer.

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

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

4. A formal verification of the solution

We consider the polynomial encoding (cod, s) from SAT in $\mathbf{\Pi}$ defined as follows: let φ be a Boolean formula in conjunctive normal form (a conjunction of clauses such that each clause is the disjunction of one or more literals) and simplified (in each clause, literals are not repeated, and also none of the clauses contains both a literal and its negation). Let $Var(\varphi) = \{x_1, \cdots, x_n\}$ be the set of propositional variables and $\{C_1, \cdots, C_p\}$ the set of clauses of φ. Let us assume that the number of variables and the number of clauses of the input formula φ, are greater than or equal to 2. Then, we define $s(\varphi) = \langle n, p\rangle$ and

$$cod(\varphi) = \{x_{i,j,0} \mid\ x_i \in C_j\} \ \cup\ \{\overline{x}_{i,j,0} \mid \neg x_i \in C_j\} \ \cup\ \{x^*_{i,j,0} \mid x_i \notin C_j, \neg x_i \notin C_j\}$$

Notice that we can represent this multiset as a matrix, in such a way that the j-th row ($1 \le j \le p$) encodes the j-th clause C_j of φ. For instance, the formula $\varphi = (x_1 + x_2 + \neg x_3)(\neg x_2 + x_4)(\neg x_2 + x_3 + \neg x_4)$ is encoded as follows:

$$cod(\varphi) = \begin{pmatrix} x_{1,1,0} & x_{2,1,0} & \overline{x}_{3,1,0} & x^*_{4,1,0}\\ x^*_{1,2,0} & \overline{x}_{2,2,0} & x^*_{3,2,0} & x_{4,2,0}\\ x^*_{1,3,0} & \overline{x}_{2,3,0} & x_{3,3,0} & \overline{x}_{4,3,0}\end{pmatrix}$$

We denote by $cod_k(\varphi)$ the multiset $cod(\varphi)$ when the third index of all objects is equal to k. For instance:

$$cod_3(\varphi) = \begin{pmatrix} x_{1,1,3} & x_{2,1,3} & \overline{x}_{3,1,3} & x^*{}_{4,1,3}\\ x^*{}_{1,2,3} & \overline{x}_{2,2,3} & x^*{}_{3,2,3} & x_{4,2,3}\\ x^*{}_{1,3,3} & \overline{x}_{2,3,3} & x_{3,3,3} & \overline{x}_{4,3,3}\end{pmatrix}$$

The Boolean formula φ will be processed by the system $\Pi(s(\varphi))$ with input multiset $cod(\varphi)$. Next, we informally describe how that system works.

The solution proposed follows a brute force algorithm in the framework of recognizer P systems with active membranes, minimal cooperation and minimal production in object evolution rules. It 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, n being the number of variables in φ.
- *Production of enough copies for each truth assignment*: in this stage p copies of each truth assignment are produced to allow the checking of the literal associated with each variable in each clause. This stage takes exactly p computation steps.
- *First Checking stage*: checking whether or not each clause of the input formula φ is satisfied by the truth assignments generated in the previous stage, encoded by each membrane labelled by 2. This stage takes exactly one computation step.

- *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, p being the number of clauses of φ.
- *Output stage*: the system sends to the environment the right answer according to the results of the previous stage. This stage takes exactly 4 steps.

4.1. Generation stage

At this stage, all the truth assignments for the variables associated with the Boolean formula φ are generated, by applying division rules from **2.1** in membranes labelled by 2. In such manner at the i-th step, $1 \le i \le n$, of this stage, division rule is triggered by object $a_{i,i}$, producing two new membranes with all its remaining contents replicated in the new membranes labelled by 2. This stage ends when objects $t_{i,n}$, $f_{i,n}$, $1 \le i \le n$, have been generated.

Proposition 1. *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_q)$ be a computation of the system $\Pi(s(\varphi))$ with input multiset $cod(\varphi)$.*

(a) For each k, $1 \le k \le n-1$, at configuration $\mathcal{C}_k$ we have $\mathcal{C}_k(1) = \{\alpha, \beta_k\}$ (being $\mathcal{C}_t(i)$ the contents of membrane i at the moment t) and there are 2^k membranes labelled by 2 such that each of them contains: the set $\{a_{i,k+1} \mid k+1 \le i \le n\}$ and the set $cod_k(\varphi)$; the multiset $\{T^p_{i,k}, F^p_{i,k} \mid 1 \le i \le n\}$; and a different subset $\{r_{1,k}, \ldots, r_{k,k}\}$, being $r \in \{t, f\}$.
(b) At configuration $\mathcal{C}_n$ we have $\mathcal{C}_n(1) = \{\alpha, \beta_n\}$ and there are 2^n membranes labelled by 2 such that each of them contains: the set $cod_n(\varphi)$; the multiset $\{T^p_i, F^p_i \mid 1 \le i \le n\}$; and a different subset $\{r_{1,n}, \ldots, r_{n,n}\}$, being $r \in \{t, f\}$.

Proof.

(a) By induction on k. The base case $k = 1$ follows bearing in mind that configuration $\mathcal{C}_1$ is obtained from configuration $\mathcal{C}_0$ by applying the rules $[\beta_0 \to \beta_1]_1$, $[a_{1,1}]_2 \to [t_{1,1}]_2\,[f_{1,1}]_2$, $[T_{i,0} \to T_{i,1}]_2$, $[F_{i,0} \to F_{i,1}]_2$, for $1 \le i \le n$, $[x_{i,j,0} \to x_{i,j,1}]_2$, $[\overline{x}_{i,j,0} \to \overline{x}_{i,j,1}]_2$, $[x^*_{i,j,0} \to x^*_{i,j,1}]_2$, for $1 \le i \le n, 1 \le j \le p$.
Let us assume that the result holds for k, $1 \le k < n-1$. Let us see that the result also holds for $k+1$.
On the one hand, at configuration $\mathcal{C}_k$ we have $\mathcal{C}_k(1) = \{\alpha, \beta_k\}$ and there are 2^k membranes labelled by 2 each of them containing the set $\{a_{i,k+1} \mid k+1 \le i \le n\}$ and the set $cod_k(\varphi)$; the multiset $\{T^p_{i,k}, F^p_{i,k} \mid 1 \le i \le n\}$; and a different subset $\{r_{1,k}, \ldots, r_{k,k}\}$, being $r \in \{t, f\}$.
On the other hand, configuration $\mathcal{C}_{k+1}$ is obtained from configuration $\mathcal{C}_k$ by applying the rules: $[\beta_k \to \beta_{k+1}]_1$, $[a_{k+1,k+1}]_2 \to [t_{k+1,k+1}]_2\,[f_{k+1,k+1}]_2$, $[t_{i,k} \to t_{i,k+1}]_2$, $[f_{i,k} \to f_{i,k+1}]_2$, for $1 \le i \le k$, $[a_{i,k+1} \to a_{i,k+2}]_2$, for $k+2 \le i \le n$, $[T_{i,k} \to T_{i,k+1}]_2$, $[F_{i,k} \to F_{i,k+1}]_2$, for $1 \le i \le n$, $[x_{i,j,k} \to x_{i,j,k+1}]_2$, $[\overline{x}_{i,j,k} \to \overline{x}_{i,j,k+1}]_2$, $[x^*_{i,j,k} \to x^*_{i,j,k+1}]_2$, for $1 \le i \le n, 1 \le j \le p$.
Hence, the result holds for $k+1$.

(b) By applying (a) to $k = n-1$ at configuration $\mathcal{C}_{n-1}$ we have $\mathcal{C}_{n-1}(1) = \{\alpha, \beta_{n-1}\}$ and there are 2^{n-1} membranes labelled by 2 each of them containing: the object $a_{n,n}$ and the set $cod_{n-1}(\varphi)$; the multiset $\{T^p_{i,k}, F^p_{i,k} \mid 1 \le i \le n\}$; and a different subset $\{r_{1,n-1}, \ldots, r_{n-1,n-1}\}$, being $r \in \{t, f\}$.
Then, (b) follows noting that configuration $\mathcal{C}_n$ is obtained from configuration $\mathcal{C}_{n-1}$ by applying the rules: $[\beta_{n-1} \to \beta_n]_1$, $[a_{n,n}]_2 \to [t_{n,n}]_2\,[f_{n,n}]_2$, $[t_{i,n-1} \to t_{i,n}]_2$, $[f_{i,n-1} \to f_{i,n}]_2$, for $1 \le i \le n-1$, $[x_{i,j,n-1} \to x_{i,j,n}]_2$, $[\overline{x}_{i,j,n-1} \to \overline{x}_{i,j,n}]_2$, $[x^*_{i,j,n-1} \to x^*_{i,j,n}]_2$, for $1 \le i \le n, 1 \le j \le p$. □

4.2. Producing enough copies for each truth assignment

At this stage, in each membrane labelled by 2, a sufficient number of copies from each truth assignment will be generated. Specifically, p copies of each of them will be produced, where p is the number of clauses of the input formula. Let us recall that in the initial configuration there are p copies of $T_1, F_1, \ldots T_n, F_n$. These copies are replicated in the 2^n membranes labelled by 2 produced by applying division rules where a copy of each truth assignment is produced. By using cooperation we use the values t_i and f_i of the truth assignment associated with each membrane labelled by 2 to remove a copy of the opposite value F_i or T_i, respectively. This stage takes exactly p steps.

Proposition 2. *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_q)$ be a computation of the system $\Pi(s(\varphi))$ with input multiset $cod(\varphi)$.*

(a) For each k, $1 \le k \le p-1$, at configuration $\mathcal{C}_{n+k}$ we have $\mathcal{C}_{n+k}(1) = \{\alpha, \beta_{n+k}\}$ and there are 2^n membranes labelled by 2 such that each of them contains: the set $cod_{n+k}(\varphi)$; a different subset $\{r_{1,n+k}, \ldots, r_{n,n+k}\}$, being $r \in \{t, f\}$; and the corresponding multiset $\{R^p_1, \overline{R}_1^{p-k}, \ldots, R^p_n, \overline{R}_n^{p-k}\}$ verifying the following: for each k, $1 \le i \le n$, if $r_{i,n+k} = t_{i,n+k}$ then $R_i = T_i$ and $\overline{R}_i = F_i$; if $r_{i,n+k} = f_{i,n+k}$ then $R_i = F_i$ and $\overline{R}_i = T_i$;
(b) At configuration $\mathcal{C}_{n+p}$ we have $\mathcal{C}_{n+p}(1) = \{\alpha, \beta_{n+p}\}$ and there are 2^n membranes labelled by 2 such that each of them contains: n copies of object #; the set $cod_{n+p}(\varphi)$; and a different subset $\{R^p_1, \ldots, R^p_n\}$ being $R \in \{T, F\}$.

Proof.

(a) By induction on k. The base case $k = 1$ follows bearing in mind that configuration $\mathcal{C}_{n+1}$ is obtained from configuration $\mathcal{C}_n$ by applying the rules: $[\beta_n \to \beta_{n+1}]_1$, $[t_{i,n}F_i \to t_{i,n+1}]_2$, $[f_{i,n}T_i \to f_{i,n+1}]_2$, for $1 \le i \le n$, $[x_{i,j,n} \to x_{i,j,n+1}]_2$, $[\overline{x}_{i,j,n} \to \overline{x}_{i,j,n+1}]_2$, $[x^*_{i,j,n} \to x^*_{i,j,n+1}]_2$, for $1 \le i \le n$, $1 \le j \le p$.
Let us assume that the result holds for k, $1 \le k < p-1$. Let us see that the result also holds for $k+1$.
On the one hand, at configuration $\mathcal{C}_{n+k}$ we have $\mathcal{C}_{n+k}(1) = \{\alpha, \beta_{n+k}\}$ and there are 2^n membranes labelled by 2 each of them containing the set $cod_{n+k}(\varphi)$; a different subset $\{r_{1,n+k}, \dots, r_{n,n+k}\}$, being $r \in \{t, f\}$; and the corresponding multiset $\{R_1^p, \overline{R}_1^{p-k}, \dots, R_n^p, \overline{R}_n^{p-k}\}$ verifying the following: for each k, $1 \le i \le n$, if $r_{i,n+k} = t_{i,n+k}$ then $R_i = T_i$ and $\overline{R}_i = F_i$; if $r_{i,n+k} = f_{i,n+k}$ then $R_i = F_i$ and $\overline{R}_i = T_i$;
On the other hand, configuration $\mathcal{C}_{n+k+1}$ is obtained from configuration $\mathcal{C}_{n+k}$ by applying the rules: $[\beta_{n+k} \to \beta_{n+k+1}]_1$, $[t_{i,n+k}F_i \to t_{i,n+k+1}]_2$, $[f_{i,n+k}T_i \to f_{i,n+k+1}]_2$, for $1 \le i \le n$, $[x_{i,j,n+k} \to x_{i,j,n+k+1}]_2$, $[\overline{x}_{i,j,n+k} \to \overline{x}_{i,j,n+k+1}]_2$, $[x^*_{i,j,n+k} \to x^*_{i,j,n+k+1}]_2$, for $1 \le i \le n$, $1 \le j \le p$.
Hence, the result holds for $k+1$.

(b) By applying (a) to $k = p-1$, at configuration $\mathcal{C}_{n+p-1}$ we have $\mathcal{C}_{n+p-1}(1) = \{\alpha, \beta_{n+p-1}\}$ and there are 2^n membranes labelled by 2 each of them containing the set $cod_{n+p-1}(\varphi)$; a different subset $\{r_{1,n+p-1}, \dots, r_{n,n+p-1}\}$, being $r \in \{t, f\}$; and the corresponding multiset $\{R_1^p, \overline{R}_1^1, \dots, R_n^p, \overline{R}_n^1\}$ verifying the following: for each i, $1 \le i \le n$, if $r_{i,n+p-1} = t_{i,n+p-1}$ then $R_i = T_i$, $\overline{R}_i = F_i$ and if $r_{i,n+p-1} = f_{i,n+p-1}$ then $R_i = F_i$, $\overline{R}_i = T_i$.
Then, (b) follows noting that configuration $\mathcal{C}_{n+p}$ is obtained from configuration $\mathcal{C}_{n+p-1}$ by applying the rules: $[\beta_{n+p-1} \to \beta_{n+p}]_1$, $[t_{i,n+p-1}F_i \to \#]_2$, $[f_{i,n+p-1}T_i \to \#]_2$, for $1 \le i \le n$, $[x_{i,j,n+p-1} \to x_{i,j,n+p}]_2$, $[\overline{x}_{i,j,n+p-1} \to \overline{x}_{i,j,n+p}]_2$, $[x^*_{i,j,n+p-1} \to x^*_{i,j,n+p}]_2$, for $1 \le i \le n$, $1 \le j \le p$. $\square$

4.3. First checking stage

At this stage, we try to determine the clauses satisfied by the truth assignments encoded by each membrane labelled by 2. For that, rules from **2.4** will be applied in such a manner that an object c_j is produced if and only if the truth assignment encoded by that membrane makes true clause C_j. This stage takes exactly one step.

Proposition 3. *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_q)$ be a computation of the system $\Pi(s(\varphi))$ with input multiset $cod(\varphi)$. At configuration $\mathcal{C}_{n+p+1}$ we have $\mathcal{C}_{n+p+1}(1) = \{\alpha\ ,\ \beta_{n+p+1}\}$ and there are 2^n membranes labelled by 2 such that each of them contains t_j copies of object c_j, for $1 \le j \le p$, if and only if the truth assignment encoded by that membrane makes true exactly t_j literals of clause C_j, and $np - (t_1 + \cdots + t_p)$ copies of object #.*

Proof. It suffices to note that configuration $\mathcal{C}_{n+p+1}$ is obtained from configuration $\mathcal{C}_{n+p}$ by applying the rules: $[\beta_{n+p} \to \beta_{n+p+1}]_1$, $[T_i x_{i,j,n+p} \to c_j]_2$, $[T_i \overline{x}_{i,j,n+p} \to \#]_2$, $[T_i x^*_{i,j,n+p} \to \#]_2$, $[F_i x_{i,j,n+p} \to \#]_2$, $[F_i \overline{x}_{i,j,n+p} \to c_j]_2$, $[F_i x^*_{i,j,n+p} \to \#]_2$, for $1 \le i \le n$, $1 \le j \le p$. $\square$

4.4. Second checking stage

At this stage, we try to determine whether some truth assignment encoded by a membrane labelled by 2 satisfies all clauses of the input formula. To that end, rules from **2.5** will be applied in such a manner that object d_j $(2 \le j \le p)$ is produced in a membrane labelled by 2 if and only if the truth assignment encoded by that membrane makes true the clauses $C_1, \dots, C_j$. Then, the input formula will be satisfied by the truth assignment encoded by a membrane labelled by 2 if and only if object d_p appears in that membrane. This stage takes exactly $p-1$ computation steps.

Proposition 4. *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_q)$ be a computation of the system $\Pi(s(\varphi))$ with input multiset $cod(\varphi)$.*

(a) For each k, $1 \le k \le p-1$, at configuration $\mathcal{C}_{(n+p+1)+k}$ we have $\mathcal{C}_{(n+p+1)+k}(1) = \{\alpha\ ,\ \beta_{(n+p+1)+k}\}$ and there are 2^n membranes labelled by 2 such that each of them contains an object d_{k+1} if and only if the truth assignment encoded in that membrane, makes true clauses $C_1, \dots, C_{k+1}$.
(b) φ is satisfiable if and only if at configuration $\mathcal{C}_{n+2p}$ there exists some membrane labelled by 2 which contains an object d_p.

Proof.

(a) By induction on k. For the base case $k = 1$ it suffices to note that configuration $\mathcal{C}_{n+p+2}$ is obtained from configuration $\mathcal{C}_{n+p+1}$ by applying the rules $[\beta_{n+p+1} \to \beta_{n+p+2}]_1$ and $[c_1c_2 \to d_2]_2$.
Let us assume that the result holds for k, $1 \le k < p-1$. Then, at configuration $\mathcal{C}_{(n+p+1)+k}$ we have $\mathcal{C}_{(n+p+1)+k}(1) = \{\alpha\ ,\ \beta_{(n+p+1)+k}\}$ and there are 2^n membranes labelled by 2 each of them containing an object d_{k+1} if and only if the truth assignment encoded in that membrane, makes true clauses $C_1, \dots, C_{k+1}$.

Bearing in mind that configuration $\mathcal{C}_{(n+p+1)+k+1}$ is obtained from configuration $\mathcal{C}_{(n+p+1)+k}$ by applying the rules $[\beta_{(n+p+1)+k} \to \beta_{(n+p+1)+k+1}]_1$ and $[d_{k+1}c_{k+2} \to d_{k+2}]_2$, we deduce that the result holds for $k+1$.

(b) In order to prove (b), let us note that the input formula φ is satisfiable if and only if there exists a truth assignment σ making true φ, that is, making true the clauses $C_1, \ldots, C_p$. From (a) we deduce that φ is satisfiable if and only at configuration $\mathcal{C}_{n+2p}$ there exists some membrane labelled by 2 which contains an object d_p. $\square$

4.5. Output stage

The output phase starts at the $(n+2p+1)$-th step, and takes exactly four steps.

- *Affirmative answer*: if the input formula φ is satisfiable then at least one of the truth assignments from a membrane with label 2 makes true all clauses. Thus, a copy of object d_p will appear in that membrane at configuration $\mathcal{C}_{n+2p}$. Then, by applying the rules $[d_p]_2 \to \gamma\,[\ ]_2$ and $[\,\beta_{n+2p} \longrightarrow \beta_{n+2p+1}\,]_1$, objects γ and β_{n+2p+1} are produced in the skin membrane. At the next step, by applying rules $[\,\alpha\,\gamma \longrightarrow \gamma'\,]_1$ and $[\,\beta_{n+2p+1} \longrightarrow \beta'\,]_1$, objects γ' and β' are produced in the skin membrane. At the step $n+2p+3$, by applying rule $[\,\gamma' \longrightarrow \gamma''\,]_1$, object γ'' is produced in the skin membrane (let us notice that object β' cannot interact with α). Finally, at the next step, by applying rule $[\,\gamma''\,]_1 \longrightarrow \texttt{yes}\,[\ \ ]_1$, object yes is sent out to the environment. Hence, the computation halts and the answer of the computation is yes.
- *Negative answer*: if the input formula φ is not satisfiable then none of the truth assignments encoded by a membrane with label 2 makes the formula φ true. Thus, object d_p does not appear in any membrane labelled by 2 in configuration $\mathcal{C}_{n+2p}$. At the step $n+2p+1$, only rule $[\,\beta_{n+2p} \longrightarrow \beta_{n+2p+1}\,]_1$ is applicable to $\mathcal{C}_{n+2p}$. Then, $\mathcal{C}_{n+2p+1}(1) = \{\alpha\,,\,\beta_{n+2p+1}\}$. At the next step, by applying rule $[\,\beta_{n+2p+1} \longrightarrow \beta'\,]_1$ we have $\mathcal{C}_{n+2p+2}(1) = \{\alpha\,,\,\beta'\}$. At the step $n+2p+3$, rule $[\,\alpha\,\beta' \longrightarrow \beta''\,]_1$ produces an object β'' in the skin membrane. Finally, at the last step, by applying rule $[\,\beta''\,]_1 \longrightarrow \texttt{no}\,[\ \ ]_1$ an object no is released to the environment. Consequently, the computation halts and the answer of the computation is no.

5. Main results

Theorem 1. SAT $\in \mathbf{PMC}_{\mathcal{DAM}^0(mcmp,+c,-d,-n)}$.

Proof. The family of P systems previously constructed verifies the following:

(a) Every system of the family $\mathbf{\Pi}$ belongs to $\mathcal{DAM}^0(mcmp, +c, -d, -n)$.

(b) The family $\mathbf{\Pi}$ is polynomially uniform by Turing machines because for each $n, p \in \mathbb{N}$, the amount of resources needed to build $\Pi(\langle n, p\rangle)$ is of a polynomial order in n and p:
- Size of the alphabet: $3np^2 + 3n^2p + 5np + \frac{7n^2}{2} + \frac{5n}{2} + 4p + 10 \in \Theta(max\{np^2, n^2p\})$.
- Initial number of membranes: $2 \in \Theta(1)$.
- Initial number of objects in membranes: $2np + n + 2 \in \Theta(np)$.
- Number of rules: $3np^2 + 3n^2p + 8np + \frac{7n^2}{2} + \frac{n}{2} + 3p + 7 \in \Theta(max\{np^2, n^2p\})$.
- Maximal number of objects involved in any rule: $3 \in \Theta(1)$.

(c) The pair (cod, s) of polynomial-time computable functions defined fulfil the following: for each input formula φ of the SAT problem, $s(\varphi)$ is a natural number, $cod(\varphi)$ is an input multiset of the system $\Pi(s(\varphi))$, and for each $n \in \mathbb{N}$, $s^{-1}(n)$ is a finite set.

(d) The family $\mathbf{\Pi}$ is polynomially bounded: indeed, for each input formula φ of the SAT problem, the deterministic P system $\Pi(s(\varphi)) + cod(\varphi)$ takes exactly $n+2p+4$ steps, n being the number of variables of φ and p the number of clauses.

(e) The family $\mathbf{\Pi}$ is sound with regard to (X, cod, s): indeed, for each input formula φ, if the computation of $\Pi(s(\varphi)) + cod(\varphi)$ is an accepting computation, then φ is satisfiable.

(f) The family $\mathbf{\Pi}$ is complete with regard to (X, cod, s): indeed, for each input formula φ such that it is satisfiable, the computation of $\Pi(s(\varphi)) + cod(\varphi)$ is an accepting computation.

Therefore, the family $\mathbf{\Pi}$ of P systems previously constructed solves the SAT problem in polynomial time in a uniform way. $\square$

Corollary 1. $\mathbf{NP} \cup \mathbf{co-NP} \subseteq \mathbf{PMC}_{\mathcal{DAM}^0(mcmp,+c,-d,-n)}$.

Proof. It suffices to note that the SAT problem is an **NP**-complete problem, SAT $\in \mathbf{PMC}_{\mathcal{DAM}^0(mcmp,+c,-d,-n)}$, and class $\mathbf{PMC}_{\mathcal{DAM}^0(mcmp,+c,-d,-n)}$ is closed under polynomial-time reduction and under complement. $\square$

Corollary 2. $\mathbf{P} = \mathbf{PMC}_{\mathcal{SAM}^0(mcmp,+c,-d,-n)}$.

Proof. It suffices to notice that $\mathbf{P} = \mathbf{PMC}_{\mathcal{SAM}^0(bmc,+c,-d,-n)}$ and each rule using minimal cooperation and minimal production is also a rule using bounded minimal cooperation, and realizing that the class is closed under polynomial-time reduction. For the reverse inclusion, we only need to keep in mind the `Sevilla theorem` to see that we can simulate any Deterministic Turing Machine with this kind of membrane systems. □

6. Conclusions

Limitations of polarizationless P systems with active membranes not using dissolution rules, with respect to efficiency, are well known. In this paper, the computational efficiency of such kind of P systems using only division rules for elementary membranes is studied in the case that a very restrictive cooperation in object evolution rules is considered. Specifically, the left-hand side of the rules consists of at most two objects and each such rule only can produce a single object. The efficiency of these systems is shown, improving a result concerning object evolution rules with minimal cooperation, where the length of their right-hand side is less than or equal to the corresponding left-hand side.

It is worth pointing out that the situation is completely different when division rules is replaced by separation rules; that is, when in the mechanism of producing an exponential number of membranes in linear time, *distribution* of objects is considered instead of the *replication* of objects. In this case, only problems in class **P** can be efficiently solved by families of polarizationless P systems with active membranes which use minimal cooperation and minimal production in object evolution rules. Consequently, new frontiers of efficiency are obtained.

These results confirm the strength of the replication with respect to the distribution of objects, from an efficiency point of view, and the irrelevant role played by dissolution when minimal cooperation is considered.

As future work, we propose to study polarizationless P systems with active membranes when cooperation in communication rules is considered instead of cooperation in object evolution rules. It seems that, in this case, division rules for non-elementary membranes can play a relevant role from a computational complexity point of view.

Acknowledgements

This work was partially supported by Grant numbers 61472328 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 Inform. 41 (2–3) (2004) 111–144.
[3] A. Alhazov, R. Freund, Gh. Păun, P systems with active membranes and two polarizations, in: Gh. Păun, A. Riscos-Núñez, A. Romero-Jiménez, F. Sancho-Caparrini (Eds.), Proceedings of the Second Brainstorming Week on Membrane Computing, Sevilla, 2–7 February 2004, Research Group on Natural Computing, TR 01/2004, Universidad de Sevilla, Fénix Editora, 2004, pp. 20–35.
[4] A. Alhazov, M.J. Pérez-Jiménez, Uniform solution of QSAT using polarizationless active membranes, in: Lecture Notes in Computer Science, vol. 4664, 2007, pp. 122–133.
[5] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, 1979.
[6] 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, pp. 137–154.
[7] 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: Lecture Notes in Computer Science, vol. 3850, 2006, pp. 224–240.
[8] L. Pan, T.-O. Ishdorj, P systems with active membranes and separation rules, J.UCS 10 (5) (2004) 630–649.
[9] L. Pan, A. Alhazov, Solving HPP and SAT by P systems with active membranes and separation rules, Acta Inform. 43 (2006) 131–145.
[10] L. Pan, A. Alhazov, T.-O. Ishdorj, Further remarks on P systems with active membranes, separation, merging, and release rules, Soft Comput. 8 (2004) 1–5.
[11] Gh. Păun, Computing with membranes, J. Comput. System Sci. 61 (1) (2000) 108–143.
[12] Gh. Păun, Attacking NP-complete problems, in: I. Antoniou, C. Calude, M.J. Dinneen (Eds.), Unconventional Models of Computation, UMC'2K, Springer-Verlag, 2000, pp. 94–115.
[13] Gh. Păun, G. Rozenberg, A. Salomaa, The Oxford Handbook of Membrane Computing, Oxford University Press, 2010.
[14] M.J. Pérez-Jiménez, F.J. Romero-Campero, Trading polarizations for bi-stable catalysts in P systems with active membranes, in: Lecture Notes in Computer Science, vol. 3365, 2005, pp. 373–388.
[15] A.E. Porreca, G. Mauri, C. Zandron, Complexity classes for membrane systems, RAIRO Theor. Inform. Appl. 40 (2) (2006) 141–162.
[16] P. Sosík, PSPACE limits the power of P systems with active membranes, J. Autom. Lang. Comb. 19 (1–4) (2014) 291–304.
[17] P. Sosík, A. Rodríguez-Patón, Membrane computing and complexity theory: a characterization of PSPACE, J. Comput. System Sci. 73 (1) (2007) 137–152.
[18] L. Valencia-Cabrera, D. Orellana-Martín, M.A. Martínez-del-Amor, A. Riscos-Núñez, M.J. Pérez-Jiménez, Polarizationless P systems with active membranes: computational complexity aspects, J. Autom. Lang. Comb. 21 (1–2) (2016) 101–117.
[19] L. Valencia-Cabrera, D. Orellana-Martín, M.A. Martínez-del-Amor, A. Riscos-Núñez, M.J. Pérez-Jiménez, Computational efficiency of minimal cooperation and distribution in polarizationless P systems with active membranes, Fund. Inform. 153 (1–2) (2017) 147–172, http://dx.doi.org/10.3233/FI-2017-1535.
[20] L. Valencia-Cabrera, D. Orellana-Martín, A. Riscos-Núñez, M.J. Pérez-Jiménez, Minimal cooperation in polarizationless P systems with active membranes, in: C. Graciani, et al. (Eds.), Proceedings of the Fourteenth Brainstorming Week on Membrane Computing, Sevilla, 1–5 February 2016, Research Group on Natural Computing, TR 01/2016, Universidad de Sevilla, Fénix Editora, 2016, pp. 327–356.