# Cell-like P systems with evolutional symport/antiport rules and membrane creation ☆

Bosheng Song [a,*], Kenli Li [a], David Orellana-Martín [b], Luis Valencia-Cabrera [b], Mario J. Pérez-Jiménez [b]

[a] College of Information Science and Engineering, Hunan University, and the National Supercomputing Center in Changsha, 410082, Hunan, China

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

*Keywords:*
Bio-inspired computing
Membrane computing
Cell-like P system
Evolutional symport/antiport rule
Membrane creation

A B S T R A C T

Cell-like P systems with symport/antiport rules are computing models inspired by the *conservation law*, in the sense that they compute by changing the places of objects with respect to the membranes, and not by changing the objects themselves. In this work, a variant of these kinds of membrane systems, called cell-like P systems with *evolutional* symport/antiport rules, where objects can evolve in the execution of such rules, is introduced. Besides, inspired by the autopoiesis process (ability of a system to maintain itself), membrane creation rules are considered as an efficient mechanism to provide an exponential workspace in terms of membranes. The presumed efficiency of these computing models (ability to solve computationally hard problems in polynomial time and uniform way) is explored. Specifically, an efficient solution to the SAT problem is provided by means of a family of recognizer cell-like P systems with evolutional symport/antiport rules and membrane creation which make use of communication rules involving a restricted number of objects.

## 1. Introduction

*Membrane Computing* is a computational paradigm which arises as an abstraction of the compartmentalized structure of living cells, and the way biochemical substances are processed in (or moved between) membrane-bounded regions [29]. The computing models in this paradigm are the so-called *P systems* or *membrane systems*. In this framework, three main approaches have been widely studied: *cell-like P systems*, organized with a hierarchical (cell-like) arrangement of membranes (a rooted tree), simulating the organization within a cell [29]; *tissue-like P systems*, inspired by the cell-interconnection communication between cells in a living tissue [22,32]; and *neural-like P systems*, inspired from the way that neurons communicate with each other by means of short electrical impulses (spikes), emitted at precise moments of time [18,19].

☆ The work of B. Song was supported by National Natural Science Foundation of China (61972138, 61602192), the Fundamental Research Funds for the Central Universities (531118010355). The work of D. Orellana-Martín, L. Valencia-Cabrera and M.J. Pérez-Jiménez is supported by the research 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 de Desarrollo Regional (FEDER) of the European Union.

* Corresponding author.

*E-mail addresses:* boshengsong@hnu.edu.cn (B. Song), lkl@hnu.edu.cn (K. Li), dorellana@us.es (D. Orellana-Martín), lvalencia@us.es (L. Valencia-Cabrera), marper@us.es (M.J. Pérez-Jiménez).

The basic ingredient of cell-like P systems is the membrane structure, which is a hierarchical arrangement of membranes embedded in a unique main membrane, called the *skin* membrane [29,30,33]. Each membrane delimits a *region* (also called *compartment*), where finite sets of symbols (corresponding to the chemicals), finite sets of evolution rules (describing the possible interactions between chemicals) and other membranes can be placed. The outside region of the skin membrane is called *environment*, where there are no rules placed in it. If a membrane does not contain other membranes, then it is called *elementary membrane*; otherwise, it is called *non-elementary membrane*. In terms of types of rules, two main approaches have been investigated. On the one hand, rewriting rules are used as a method to make the objects of a system to evolve and move them through the compartments of itself. P systems with active membranes are a framework where these kinds of rules have defined several frontiers of efficiency in the framework of Membrane Computing. On the other hand, symport/antiport rules can be seen as an abstraction of the movement of chemical compounds between different compartments of a system [6,7,27,28], let it be a single cell or a living tissue. In fact, as it happens in living systems, this communication is not limited to the compartments of the system, but they can interact with the *environment* too.

Membrane division and membrane separation are inspired by the mitosis and the membrane fission processes, which provide a mechanism to generate an exponential workspace in linear time. With the help of this mechanism, various P systems have manifested an excellent performance with respect to address computationally hard problems, including **NP**-complete problems [10,11,32,36,44], **PSPACE**-complete problems [2–4,45,46].

Membrane creation is another mechanism to produce new membranes, which was first considered in [23], inspired from the fact that when a membrane compartment becomes too large, new membranes often appear inside it. Membrane creation differs from membrane division and membrane separation, since no new membranes with new labels are created by means of the execution of a division rule or a separation rule, they only duplicate membranes while maintaining their labels. While using creation rules, we have to take care of how many labels will be used and where can they be used. As expected, P systems with membrane creation can solve **NP**-complete problem (subset sum) [15], even **PSPACE**-complete problem (QSAT) [16] in a polynomial time.

In [43], this last approach was extended to tissue P systems in such a way that, while classical symport/antiport rules only transport objects between the different compartments [1,5,8,12], evolutional symport/antiport rules are capable of modifying them while they travel to a different compartment. As a new methodology to tackle the **P** *versus* **NP** problem [13,14,20,21], new frontiers between the *non efficiency* (only problems in class **P** can be efficiently solved) and the *presumed efficiency* (ability to solve computationally hard problems in polynomial time and uniform way) have been reached by means of these kinds of membrane systems. While using division rules, passing from rules of length at most $(1, n)$ to $(2, n)$ (the length of an evolutional symport/antiport rule is an ordered pair whose first component is the total number of objects involved in the left hand side of the rule, and the second component is the total number of objects involved in the right hand side of the rule), for every natural number $n \geq 1$, amounts to passing from the non efficiency to the presumed efficiency. If separation rules are used instead of division rules, the non efficiency of tissue P systems with evolutional symport/antiport rules of length at most $(1, n)$ or $(n, 1)$, for every natural number $n \geq 1$, has been established in [25].

In this work, we apply evolutional symport/antiport rules in cell-like P systems, and using membrane creation rules as a mechanism to construct an exponential workspace in terms of membranes in polynomial time. More precisely, an efficient solution to the SAT problem is provided by means of a family of recognizer cell-like P systems with evolutional symport/antiport rules and membrane creation which make use of communication rules involving the left-hand side (LHS) of length at most 2 and the right-hand side (RHS) of length at most 2.

We emphasize the following facts that in [17], the SAT problem was solved by P systems with membrane creation using rules of length at most $(1, 2)$, where object evolution rules, communication rules, creation rules and *dissolution rules* are applied; however, in this work, the SAT problem can be efficiently solved by P systems with evolutional symport/antiport rules and membrane creation using rules of length at most $(2, 2)$, where evolutional communication rules and creation rules are used. Although to a certain extent, evolutional communication rules, evolution rules and communication rules can mutual transform, "dissolution rules" are a very power tool to achieve the desired computational power as described in [14,17], it is shown that dissolution rules provide a borderline between efficiency and non-efficiency for P systems with membrane creation, that is, P systems with membrane creation without dissolution rules characterize standard class **P** [14]. In the present work, dissolution rules are *not* allowed for P systems with evolutional symport/antiport rules and membrane creation, and we show that **NP**-complete problem can be efficiently solved by such kind of P systems. Moreover, in P systems with membrane creation, all rules are noncooperative, that is, the length of left-hand side of rules is at most 1; however, in P systems with evolutional symport/antiport rules and membrane creation, the length of left-hand side of rules can be arbitrary. Hence there exist essential differences between P systems with membrane creation and P systems with evolutional symport/antiport rules and membrane creation.

The paper is organized as follows. In the next section, some notations and the computing model of cell-like P system with evolutional symport/antiport rules and membrane creation are presented. In section 3, the result about the computational efficiency of cell-like P systems with evolutional symport/antiport rules and membrane creation is demonstrated. Finally, some conclusions and open problems are presented.

## 2. Preliminaries and model description

Some basic notions used in this work from formal language theory are recalled, one is referred to [33,37] for further information.

An *alphabet* is denoted by $\Gamma$, which is a non-empty set, and the elements in $\Gamma$ are called *symbols*. A *string* $u$ over $\Gamma$ is a finite sequence of symbols from $\Gamma$, and the *length* of the string $u$ (denoted by $|u|$) from $\Gamma$ is the number of occurrences in $u$ of symbols. For an alphabet $\Gamma$, a *multiset* over $\Gamma$ is a pair $(\Gamma, f)$ where $f : \Gamma \to \mathbb{N}$ is a mapping, $\mathbb{N}$ is the set of natural numbers. Let $m_1$, $m_2$ be multisets over $\Gamma$. The union of $m_1$ and $m_2$, denoted by $m_1 + m_2$, is the multiset over $\Gamma$ defined as $(m_1 + m_2)(x) = m_1(x) + m_2(x)$ for each $x \in \Gamma$. The relative complement of $m_2$ in $m_1$, denoted by $m_1 \setminus m_2$, is the multiset defined as $(m_1 \setminus m_2)(x) = m_1(x) - m_2(x)$ if $m_1(x) \geq m_2(x)$, and $(m_1 \setminus m_2)(x) = 0$ otherwise. We denote by $\emptyset$ the empty multiset and by $M_f(\Gamma)$ the set of all finite multisets over $\Gamma$.

Next we give the definition of cell-like P systems with evolutional symport/antiport rules and membrane creation.

**Definition 1.** A cell-like P system with evolutional symport/antiport rules and membrane creation of degree $q \geq 1$ is a tuple

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

where

- $\Gamma$ is a finite alphabet of objects;
- $\mathcal{E}$ is a finite alphabet of objects initially placed in the environment (note that each object initially placed in the environment is available in an arbitrary number of copies), such that $\mathcal{E} \subseteq \Gamma$;
- $H$ is a finite alphabet such that $\{0, 1, \ldots, q\} \subseteq H$ (the environment is identified by the label 0);
- $\mu$ is a membrane structure (a rooted tree) whose nodes are bijectively labelled with $1, \ldots, q$ (the root of the tree is labelled by 1);
- $\mathcal{M}_i$, $1 \leq i \leq q$, are finite multisets over $\Gamma \setminus \mathcal{E}$;
- $\mathcal{R}$ is a finite set of rules of the following forms:
  - Evolutional symport rules:
    - Among membranes:
      - $\star$ $[\ u[\ ]_i\ ]_j \to [\ [\ u'\ ]_i\ ]_j$, where $i, j \in H$, $i \neq j$, $u, u' \in M_f(\Gamma)$, $|u| > 0$ (send-in rules);
      - $\star$ $[\ [\ u\ ]_i\ ]_j \to [\ u'[\ ]_i\ ]_j$, where $i, j \in H$, $i \neq j$, $u, u' \in M_f(\Gamma)$, $|u| > 0$ (send-out rules);
    - Between the root and the environment:
      - $\star$ $[\ u[\ ]_1\ ]_0 \to [\ [\ u'\ ]_1\ ]_0$, where $u, u' \in M_f(\Gamma)$, $|u| > 0$, and there exists at least one object $a \in u$, such that $a \in \Gamma \setminus \mathcal{E}$ (send-in rules);
      - $\star$ $[\ [\ u\ ]_1\ ]_0 \to [\ u'[\ ]_1\ ]_0$, where $u, u' \in M_f(\Gamma)$, $|u| > 0$ (send-out rules);
  - Evolutional antiport rules: $[\ u[\ v\ ]_i\ ]_j \to [\ v'[\ u'\ ]_i\ ]_j$, where $i, j \in H \cup \{0\}$, $i \neq j$, $u, v, u', v' \in M_f(\Gamma)$, $|u| > 0$, $|v| > 0$;
  - Creation rules: $[\ a \to [\ u\ ]_i\ ]_j$, where $i, j \in H$, $i \neq j$, $a \in \Gamma$, $u \in M_f(\Gamma)$;
- $i_{out}$ is the output region.

A *configuration* $\mathcal{C}_t$ at an instant $t$ of cell-like P system with evolutional symport/antiport rules and membrane creation at any moment is described by the current membrane structure, together with all multisets of objects over $\Gamma$ associated with the regions of this membrane structure and the multiset of objects over $\Gamma \setminus \mathcal{E}$ associated with the environment at that moment.

An evolutional send-in rule $[\ u[\ ]_i\ ]_j \to [\ [\ u'\ ]_i\ ]_j$ is applicable to a configuration $\mathcal{C}_t$ if there exists a region $i$ from $\mathcal{C}_t$ whose parent membrane, labelled by $j$, contains multiset $u$. When applying such a rule, the multiset of objects $u$ in region $j$ is consumed and the multiset of objects $u'$ is produced in region $i$ from $\mathcal{C}_{t+1}$. An evolutional sent-out rule $[\ [\ u\ ]_i\ ]_j \to [\ u'[\ ]_i\ ]_j$ is applicable to a configuration $\mathcal{C}_t$ if there is a region $i$ from $\mathcal{C}_t$ which contains multiset $u$ and the parent of $i$ is labelled by $j$. When applying such a rule, the multiset of objects $u$ in region $i$ from $\mathcal{C}_t$ is consumed and the multiset of objects $u'$ is generated in region $j$ from $\mathcal{C}_{t+1}$.

An evolutional antiport rule $[\ u[\ v\ ]_i\ ]_j \to [\ v'[\ u'\ ]_i\ ]_j$ is applicable to a configuration $\mathcal{C}_t$ if there exists a membrane $i$ from $\mathcal{C}_t$ which contains multiset $v$, whose parent region is labelled by $j$ and contains multiset $u$. When applying such a rule, the multiset of objects $v$ in membrane $i$ and the multiset of objects $u$ in region $j$ are consumed; simultaneously, the multiset of objects $u'$ is produced in membrane $i$ from $\mathcal{C}_{t+1}$ and the multiset of objects $v'$ is produced in region $j$.

A creation rule $[\ a \to [\ u\ ]_i\ ]_j$ is applicable to a configuration $\mathcal{C}_t$ if there exists a membrane $j$ from $\mathcal{C}_t$ which contains object $a$. When applying such a rule, under the influence of object $a$, a new membrane with label $i$ having inside the multiset $u$, is created in such manner that it will be a daughter of the membrane with label $j$.

Following the definition in [25], in this work, we denote by $\mathcal{CCEC}(k_1, k_2)$ ($k_1, k_2$ are natural numbers, and $k_1 \geq 1$, $k_2 \geq 1$) the class of cell-like P systems with evolutional symport/antiport rules and membrane creation which makes use of evolutional communication rules such that the total number of objects involved in the LHS is at most $k_1$ and the total number of objects involved in the RHS is at most $k_2$.

The rules of a cell-like P system with evolutional symport/antiport rules and membrane creation are used in the following manner: at each step, a maximal multiset of rules is applied (no further rule can be added being applicable) with the

following restriction: when a creation rule is applied to a membrane $j$, this rule is the only one which is applied for that membrane at that step, that is, the objects inside that membrane do not evolve by means of communication rules. The objects in the new membranes resulting from creation could participate in the interaction with the objects in the (upper or lower) neighbor of membranes by means of communication rules at the next step.

The *transition* of system $\Pi$ is defined by transferring from a configuration to a next configuration. The *computation* of $\Pi$ over the initial configuration is a finite or infinite sequence of transitions. When system $\Pi$ runs to a configuration where no rule of the system is applicable in this configuration, the system is said to run to the *halting* configuration. Only a computation reaching a halting configuration gives a result, which is encoded by the specific objects present in the output region $i_{out}$.

In order to solve decision problems, a recognizer cell-like P system with evolutional symport/antiport rules and membrane creation is given.

**Definition 2.** A recognizer cell-like P system with evolutional symport/antiport rules and membrane creation of degree $q \geq 1$ is a tuple

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

where

- $(\Gamma, \mathcal{E}, H, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$ is a cell-like P system with evolutional symport/antiport rules and membrane creation of degree $m \geq 1$, such that $\Gamma$ has two distinguished objects `yes` and `no`, $\mathcal{M}_1, \ldots, \mathcal{M}_q$ are finite multisets over $\Gamma \setminus \Sigma$;
- $\Sigma$ is an input alphabet strictly contained in $\Gamma$;
- $i_{in} \in \{1, \ldots, q\}$ is the input membrane, and $i_{out} = 0$;
- all computations halt;
- for each computation of $\Pi$, either object `yes` or object `no` (but not both) is released into the environment, at the last step of the computation.

We denote by $\mathcal{CCEC}(k_1, k_2)$ the class of all recognizer cell-like P system with evolutional symport/antiport rules and membrane creation such that the total number of objects involved in the LHS of the evolutional communication rules is at most $k_1$ and the total number of objects involved in the RHS of the evolutional communication rule is at most $k_2$.

Next, following [34,35], the concept of solving a decision problem in a uniform way and polynomial time by means of a family of recognizer membrane systems, is defined.

**Definition 3.** A decision problem $X = (I_X, \theta_X)$ is solvable in polynomial time and uniform way by a family $\mathbf{\Pi} = \{\Pi(n) \mid n \in \mathbb{N}\}$ of membrane systems from $\mathcal{CCEC}(k_1, k_2)$ if the following conditions hold:

- the family $\mathbf{\Pi}$ is polynomially uniform by Turing machines, that is, there exists a deterministic Turing machine working in polynomial time which constructs the system $\Pi(n)$ from $n \in \mathbb{N}$;
- 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 $\Pi(s(u))$; we denote by $\Pi(s(u)) + cod(u)$ the system obtained by adding $cod(u)$ to the multiset in the region $i_{in}$ of $\Pi(s(u))$;
  - 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 $\Pi(s(u))$ with input $cod(u)$ 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 $\Pi(s(u))$ with input $cod(u)$, 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 $\Pi(s(u))$ with input $cod(u)$ is an accepting one.

We denote by $\mathbf{PMC}_{\mathcal{CCEC}(k_1,k_2)}$ the set of all decision problems which can be solved in a uniform way and polynomial time by means of recognizer cell-like P system with evolutional symport/antiport rules and membrane creation, where the evolutional communication rules of length at most $(k_1, k_2)$.

## 3. An efficient solution to the SAT problem in $\mathcal{CCEC}(2, 2)$

In this section the presumed efficiency of $\mathcal{CCEC}(2, 2)$ is established, by providing a polynomial-time uniform solution to the `SAT` problem (one is referred to [24] for the detail of the `SAT` problem) by means of a family of membrane systems $\mathbf{\Pi} = \{\Pi(t) \mid t \in \mathbb{N}\}$ from $\mathcal{CCEC}(2, 2)$.

For each pair of natural numbers $n, p \in \mathbb{N}$, we consider the recognizer tissue P system from $\mathcal{CCEC}(2, 2)$, $\Pi(\langle n, p\rangle) = (\Gamma, \mathcal{E}, \Sigma, H, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{in}, i_{out})$, defined as follows:

- The working alphabet:

$$\begin{aligned}\Gamma \;=\;& \Sigma \cup \{x_{i,j,t}, x_{i,j,f}, \bar{x}_{i,j,t}, \bar{x}_{i,j,f} \mid 1 \le i \le n, 1 \le j \le p\} \cup \\ & \{a_{i,t}, a_{i,f} \mid 1 \le i \le n+1\} \cup \{c_{j,t}, c'_{j,t}, c_{j,f}, c_j, E_j, E'_j \mid 1 \le j \le p\} \cup \\ & \{\alpha_i, \alpha'_i \mid 0 \le i \le 4n+2p+5\} \cup \{a_1, b, E_{p+1}, t, \mathtt{yes}, \mathtt{no}\}.\end{aligned}$$

- The input alphabet: $\Sigma = \{x_{i,j}, \bar{x}_{i,j} \mid 1 \le i \le n, 1 \le j \le p\}$.
- The environment alphabet: $\mathcal{E} = \{\alpha'_i \mid 0 \le i \le 4n+2p+4\}$.
- The set of labels: $H = \{lt, lf \mid 1 \le l \le n\} \cup \{0, 1, 2, n+1\}$.
- The membrane structure: $\mu = [\,[\,\,]_2\,]_1$.
- The initial multisets: $\mathcal{M}_1 = \{a_1, b, \alpha_0\}$, $\mathcal{M}_2 = \emptyset$.
- $i_{in} = 1$ is the input membrane.
- $i_{out} = 0$ is the output region.
- The set $\mathcal{R}$ consists of the following rules:
  - Rules to generate the $2^n$ possible truth assignments:
    $r_1 \equiv [\,a_1[\,\,]_2\,]_1 \to [\,[\,a_1\,]_2\,]_1$,
    $r_{2,i,j} \equiv [\,x_{i,j}[\,\,]_2\,]_1 \to [\,[\,x_{i,j}\,]_2\,]_1$, $1 \le i \le n, 1 \le j \le p$,
    $r_{3,i,j} \equiv [\,\bar{x}_{i,j}[\,\,]_2\,]_1 \to [\,[\,\bar{x}_{i,j}\,]_2\,]_1$, $1 \le i \le n, 1 \le j \le p$,
    $r_4 \equiv [\,[\,a_1\,]_2\,]_1 \to [\,a_{1,t}a_{1,f}[\,\,]_2\,]_1$,
    $r_{5,i,j} \equiv [\,[\,x_{i,j}\,]_2\,]_1 \to [\,x_{i,j,t}x_{i,j,f}[\,\,]_2\,]_1$, $1 \le i \le n, 1 \le j \le p$,
    $r_{6,i,j} \equiv [\,[\,\bar{x}_{i,j}\,]_2\,]_1 \to [\,\bar{x}_{i,j,t}\bar{x}_{i,j,f}[\,\,]_2\,]_1$, $1 \le i \le n, 1 \le j \le p$,
    $r_7 \equiv [\,a_{1,t} \to [\,a_{2,t}a_{2,f}\,]_{1t}\,]_1$,
    $r_8 \equiv [\,a_{1,f} \to [\,a_{2,t}a_{2,f}\,]_{1f}\,]_1$,
    $r_{9,i,l} \equiv [\,a_{i+1,t} \to [\,a_{i+2,t}a_{i+2,f}\,]_{(i+1)t}\,]_{il}$, $1 \le i \le n-1, l = t, f$,
    $r_{10,i,l} \equiv [\,a_{i+1,f} \to [\,a_{i+2,t}a_{i+2,f}\,]_{(i+1)f}\,]_{il}$, $1 \le i \le n-1, l = t, f$,
    $r_{11,j} \equiv [\,x_{1,j,t}[\,\,]_{1t}\,]_1 \to [\,[\,c_{j,t}c_{j,f}\,]_{1t}\,]_1$, $1 \le j \le p$,
    $r_{12,j} \equiv [\,\bar{x}_{1,j,t}[\,\,]_{1t}\,]_1 \to [\,[\,\lambda\,]_{1t}\,]_1$, $1 \le j \le p$,
    $r_{13,i,j} \equiv [\,x_{i,j,t}[\,\,]_{1t}\,]_1 \to [\,[\,x_{i,j,t}x_{i,j,f}\,]_{1t}\,]_1$, $2 \le i \le n, 1 \le j \le p$,
    $r_{14,i,j} \equiv [\,\bar{x}_{i,j,t}[\,\,]_{1t}\,]_1 \to [\,[\,\bar{x}_{i,j,t}\bar{x}_{i,j,f}\,]_{1t}\,]_1$, $2 \le i \le n, 1 \le j \le p$,
    $r_{15,j} \equiv [\,x_{1,j,f}[\,\,]_{1f}\,]_1 \to [\,[\,\lambda\,]_{1f}\,]_1$, $1 \le j \le p$,
    $r_{16,j} \equiv [\,\bar{x}_{1,j,f}[\,\,]_{1f}\,]_1 \to [\,[\,c_{j,t}c_{j,f}\,]_{1f}\,]_1$, $1 \le j \le p$,
    $r_{17,i,j} \equiv [\,x_{i,j,f}[\,\,]_{1f}\,]_1 \to [\,[\,x_{i,j,t}x_{i,j,f}\,]_{1f}\,]_1$, $2 \le i \le n, 1 \le j \le p$,
    $r_{18,i,j} \equiv [\,\bar{x}_{i,j,f}[\,\,]_{1f}\,]_1 \to [\,[\,\bar{x}_{i,j,t}\bar{x}_{i,j,f}\,]_{1f}\,]_1$, $2 \le i \le n, 1 \le j \le p$,
    $r_{19,i,j,l} \equiv [\,c_{j,t}[\,\,]_{(i+1)t}\,]_{il} \to [\,[\,c_{j,t}c_{j,f}\,]_{(i+1)t}\,]_{il}$, $1 \le i \le n-1, 1 \le j \le p, l = t, f$,
    $r_{20,i,j,l} \equiv [\,c_{j,f}[\,\,]_{(i+1)f}\,]_{il} \to [\,[\,c_{j,t}c_{j,f}\,]_{(i+1)f}\,]_{il}$, $1 \le i \le n-1, 1 \le j \le p, l = t, f$,
    $r_{21,i,j,l} \equiv [\,x_{i+1,j,t}[\,\,]_{(i+1)t}\,]_{il} \to [\,[\,c_{j,t}c_{j,f}\,]_{(i+1)t}\,]_{il}$, $1 \le i \le n-1, 1 \le j \le p, l = t, f$,
    $r_{22,i,j,l} \equiv [\,\bar{x}_{i+1,j,t}[\,\,]_{(i+1)t}\,]_{il} \to [\,[\,\lambda\,]_{(i+1)t}\,]_{il}$, $1 \le i \le n-1, 1 \le j \le p, l = t, f$,
    $r_{23,i,j,l} \equiv [\,x_{i+1,j,f}[\,\,]_{(i+1)f}\,]_{il} \to [\,[\,\lambda\,]_{(i+1)f}\,]_{il}$, $1 \le i \le n-1, 1 \le j \le p, l = t, f$,
    $r_{24,i,j,l} \equiv [\,\bar{x}_{i+1,j,f}[\,\,]_{(i+1)f}\,]_{il} \to [\,[\,c_{j,t}c_{j,f}\,]_{(i+1)f}\,]_{il}$, $1 \le i \le n-1, 1 \le j \le p, l = t, f$,
    $r_{25,i,j,k,l} \equiv [\,x_{k,j,t}[\,\,]_{(i+1)t}\,]_{il} \to [\,[\,x_{k,j,t}x_{k,j,f}\,]_{(i+1)t}\,]_{il}$, $1 \le i \le n-2, 1 \le j \le p, i+2 \le k \le n, l = t, f$,
    $r_{26,i,j,k,l} \equiv [\,\bar{x}_{k,j,t}[\,\,]_{(i+1)t}\,]_{il} \to [\,[\,\bar{x}_{k,j,t}\bar{x}_{k,j,f}\,]_{(i+1)t}\,]_{il}$, $1 \le i \le n-2, 1 \le j \le p, i+2 \le k \le n, l = t, f$,
    $r_{27,i,j,k,l} \equiv [\,x_{k,j,f}[\,\,]_{(i+1)f}\,]_{il} \to [\,[\,x_{k,j,t}x_{k,j,f}\,]_{(i+1)f}\,]_{il}$, $1 \le i \le n-2, 1 \le j \le p, i+2 \le k \le n, l = t, f$,
    $r_{28,i,j,k,l} \equiv [\,\bar{x}_{k,j,f}[\,\,]_{(i+1)f}\,]_{il} \to [\,[\,\bar{x}_{k,j,t}\bar{x}_{k,j,f}\,]_{(i+1)f}\,]_{il}$, $1 \le i \le n-2, 1 \le j \le p, i+2 \le k \le n, l = t, f$,
    $r_{29,l} \equiv [\,a_{n+1,t} \to [\,E_1\,]_{n+1}\,]_{nl}$, $l = t, f$.
  - Rules to check if a given truth assignment makes true the input formula $\varphi$:
    $r_{30,j,l} \equiv [\,c_{j,t}[\,\,]_{n+1}\,]_{nl} \to [\,[\,c'_{j,t}\,]_{n+1}\,]_{nl}$, $1 \le j \le p, l = t, f$,
    $r_{31,j,l} \equiv [\,[\,c'_{j,t}\,]_{n+1}\,]_{nl} \to [\,c_j[\,\,]_{n+1}\,]_{nl}$, $1 \le j \le p, l = t, f$,
    $r_{32,j,l} \equiv [\,c_j[\,E_j\,]_{n+1}\,]_{nl} \to [\,E'_j[\,\,]_{n+1}\,]_{nl}$, $1 \le j \le p, l = t, f$,
    $r_{33,j,l} \equiv [\,E'_j[\,\,]_{n+1}\,]_{nl} \to [\,[\,E_{j+1}\,]_{n+1}\,]_{nl}$, $1 \le j \le p, l = t, f$.
  - Rules to return the correct answer:
    $r_{34,l} \equiv [\,[\,E_{p+1}\,]_{n+1}\,]_{nl} \to [\,t[\,\,]_{n+1}\,]_{nl}$, $l = t, f$,
    $r_{35,i,l,l'} \equiv [\,[\,t\,]_{il'}\,]_{(i-1)l} \to [\,t[\,\,]_{il'}\,]_{(i-1)l}$, $2 \le i \le n, l = t, f, l' = t, f$,
    $r_{36,l} \equiv [\,[\,t\,]_{1l}\,]_1 \to [\,t[\,\,]_{1l}\,]_1$, $l = t, f$,
    $r_{37} \equiv [\,[\,bt\,]_1\,]_0 \to [\,\mathtt{yes}[\,\,]_1\,]_0$,
    $r_{38,i} \equiv [\,\alpha'_i[\,\alpha_i\,]_1\,]_0 \to [\,[\,\alpha_{i+1}\,]_1\,]_0$, $0 \le i \le 4n+2p+4$,
    $r_{39} \equiv [\,[\,b\alpha_{4n+2p+5}\,]_1\,]_0 \to [\,\mathtt{no}[\,\,]_1\,]_0$.

### *3.1. An overview of the computation*

Let us consider the polynomial encoding $(cod, s)$ of the SAT problem in the family $\mathbf{\Pi}$ defined as follows: for each Boolean formula in CNF and simplified form with $n$ variables and $p$ clauses, $\varphi = C_1 \vee \cdots \vee C_p$, where $C_j = l^j_1 \wedge \cdots \wedge l^j_{m_j}$, $1 \le j \le p$ and

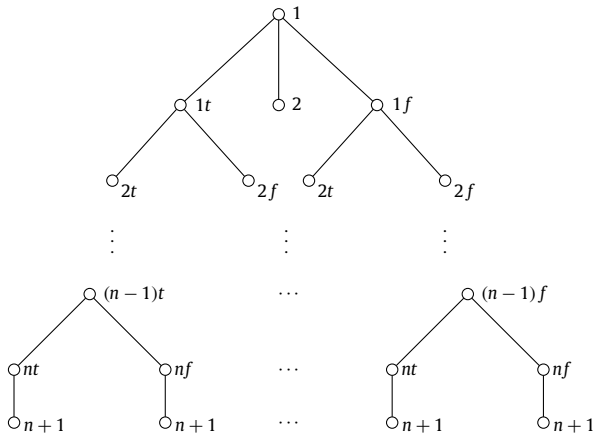


**Fig. 1.** The membrane structure of the system when the generation stage completes. Numbers at nodes indicate labels of membranes.

$l_r^j \in \{x_i, \neg x_i\}, 1 \leq i \leq n, 1 \leq r \leq m_j$, we define $s(\varphi) = \langle n, p \rangle$ and $cod(\varphi) = \{x_{i,j} \mid x_i \in C_j\} \cup \{\overline{x}_{i,j} \mid \neg x_i \in C_j\}$. Then, the system $\Pi(s(\varphi))$ with input multiset $cod(\varphi)$ (denoted by $\Pi(s(\varphi)) + cod(\varphi)$) will process the formula $\varphi$, and $\Pi(s(\varphi)) \in \mathcal{CCEC}(2, 2)$.

The system $\Pi(s(\varphi)) + cod(\varphi)$ has been designed following a brute-force algorithm and it consists on the following stages:

(a) *Generation stage*: by using creation rules all truth assignments associated with the set of variables $\{x_1, \ldots, x_n\}$ will be generated and, simultaneously, the clauses satisfied for each truth assignment are checked.
(b) *Checking stage*: truth assignment satisfying all clause of $\varphi$ are checked.
(c) *Output stage*: the system sends to the environment an object `yes` or an object `no` at the last step of the computation depending on if the formula $\varphi$ is or not satisfiable, according to the previous stage.

Next, an exhaustive description of each stage is given.

**Generation stage**

In this stage, by using membrane creation rules, a binary tree structure with $2^n$ leaves is produced, all truth assignments for the variables associated with $\varphi(x_1, \ldots, x_n)$ will be generated (corresponding to the labels of membranes). During this process, the clauses that are satisfied by the corresponding truth assignment are checked. In this way, after completing this stage, there are some of the objects $c_{1,t}, c_{1,f}, c_{2,t}, c_{2,f}, \ldots, c_{p,t}, c_{p,f}$ in a membrane with label $nt$ or $nf$, which correspond to the clauses satisfied by the truth assignment. Besides, object $\alpha_i$ in membrane 1 is evolved to count the number of steps.

In the initial configuration of the system, we have objects $a_1, b, \alpha_0, cod(\varphi)$ in membrane 1, objects $\alpha_i'$ $(0 \leq i \leq 4n+2p+4)$ in the environment.

The processes of assigning truth-assignment of variable $x_i$ as well as looking for the clauses satisfied by the truth-assignment of variable $x_i$ are described as follows. Note that from step 1 to step $4n + 2p + 5$, the count object $\alpha_i$ is evolved in each step in membrane 1; that is, the subscript of $\alpha_i$ is increased by one for each step.

At step 1, rule $r_1$ is used, object $a_1$ is sent into membrane 2; simultaneously, all the input objects $cod(\varphi)$ are sent into membrane 2 by using rules $r_{2,i,j}, r_{3,i,j}$. At step 2, rules $r_4, r_{5,i,j}, r_{6,i,j}$ are enabled and applied, all objects $a_1, x_{i,j}, \bar{x}_{i,j}$ in membrane 2 are sent to membrane 1 and evolved to $a_{1,t}a_{1,f}, x_{i,j,t}x_{i,j,f}, \bar{x}_{i,j,t}\bar{x}_{i,j,f}$, respectively. With object $a_{1,t}$ (resp., $a_{1,f}$) appears in membrane 1, rule $r_7$ (resp., $r_8$) is applied, object $a_{1,t}$ (resp., $a_{1,f}$) creates a membrane with label $1t$ (resp., $1f$), where objects $a_{2,t}, a_{2,f}$ are placed.

From step 3, rules from $r_{9,i,l}$ to $r_{29,l}$ are selected in a non-deterministic way as follows: as no communication rules can be applied to a membrane that will generate a new membrane within it in the next configuration, then three sets of rules can be applied to a single membrane in a given step: (a) rule $r_{9,i,l}$, that creates a new membrane labelled by $(i+1)t$ and two new objects $a_{i+2,t}, a_{i+2,f}$ within it by consuming one object $a_{i+1,t}$ or rule $r_{29,l}$ to create a membrane labelled by $n+1$ with an object $E_1$ to start the next stage; (b) rule $r_{10,i,l}$, that creates a new membrane labelled by $(i+1)f$ and two new objects $a_{i+2,t}, a_{i+2,f}$ within it by consuming one object $a_{i+1,f}$; and (c) rules $r_{11,j}, r_{12,j}, r_{15,j}, r_{16,j}, r_{21,i,j,l}, r_{22,i,j,l}, r_{23,i,j,l}, r_{24,i,j,l}$, devoted to check whether a given truth assignment makes a literal satisfy its clause, $r_{13,i,j}$, $r_{14,i,j}$, $r_{17,i,j}$, $r_{18,i,j}$, $r_{25,i,j,k,l}$, $r_{26,i,j,k,l}$, $r_{27,i,j,k,l}$, $r_{28,i,j,k,l}$, devoted to move literals from the parent membrane to a child membrane, $r_{19,i,j,l}, r_{20,i,j,l}$, devoted to duplicate objects $c_j$ to send them to the inner membranes.

Hence, after $3n + 3$ the generation stage completes, a binary tree structure with $2^n$ leaves is produced (see Fig. 1), where each membrane with label $n + 1$ contains an object $E_1$, each membrane with label $nt$ or $nf$ contains some objects from the set $\{c_{1,t}, c_{1,f}, c_{2,t}, c_{2,f}, \ldots, c_{p,t}, c_{p,f}\}$.

**Checking stage.**

When the generation stage completes, the system has $2^n$ copies of membrane $nt$ or $nf$, and each of these membranes contains a membrane $n + 1$ and some objects from the set $\{c_{1,t}, c_{1,f}, c_{2,t}, c_{2,f}, \ldots, c_{p,t}, c_{p,f}\}$, which correspond to clauses

satisfied by the truth assignment of the variables (note that objects $c_{j,t}, c_{j,f}$ appear in pairs). If there exists at least one membrane $nt$ or $nf$ that contains all the objects $c_{1,t}, c_{2,t}, \ldots, c_{p,t}$, then the formula $\varphi$ is satisfied by the corresponding truth assignment in that membrane; if there is no membrane $nt$ or $nf$ that contains all the objects $c_{1,t}, c_{2,t}, \ldots, c_{p,t}$, then the formula $\varphi$ is not satisfied.

The checking stage takes $2p+2$ steps, which consists of a loop with $p$ iterations, each iteration takes two steps, and two addition steps.

By using rules $r_{30,j,l}, r_{31,j,l}$, all the objects $c_{j,t}$ $(1 \leq j \leq p)$ evolve to $c_j$.

At the first step of the $j$-th loop $(1 \leq j \leq p)$, rule $r_{32,j,l}$ is enabled if and only if membrane $nt$ or $nf$ encodes a truth assignment making clauses $C_1, \ldots, C_j$ true. We remark that for any membrane $nt$ or $nf$, if it contains a truth assignment that does not make the clause $C_j$ true, then the computation will halt in that membrane. By using rule $r_{32,j,l}$, object $c_j$ is sent into membrane $n+1$ and consumed; simultaneously, object $E_j$ is sent out of membrane $n+1$ and evolved to $E'_j$.

At the second step of the $j$-th loop $(1 \leq j \leq p)$, if a membrane $nt$ or $nf$ contains object $E'_j$, then rule $r_{33,j,l}$ is enabled and applied, object $E'_j$ is sent into membrane $n+1$ and evolves to $E_{j+1}$.

**Output stage.**

If the formula $\varphi$ is satisfiable, then there exists at least one membrane $n+1$ that contains object $E_{p+1}$ after $3n+2p+3$ steps. In this case, at step $3n+2p+4$, by using rule $r_{34,l}$, object $E_{p+1}$ is sent out of membrane $n+1$ and evolved to $t$. At the next $n$ steps, by applying rules $r_{35,i,l,l'}, r_{36,l}$, object $t$ will be sent to membrane 1. At step $4n+2p+5$, rule $r_{37}$ is enabled and used, objects $b, t$ are sent to the environment and evolved to `yes`. Note that rule $r_{38,i}$ is used from step 1 to step $4n+2p+5$. Hence the computation of the system halts, the answer of the system being *affirmative*.

If the formula $\varphi$ is not satisfiable, then there is no membrane $n+1$ that contains object $E_{p+1}$ after $3n+2p+3$ steps. In this case, from step $3n+2p+4$ to step $4n+2p+5$, only rule $r_{38,i}$ is enabled and applied, so that object $\alpha_{4n+2p+5}$ will appear in membrane 1. At step $4n+2p+6$, rule $r_{39}$ is applied, objects $\alpha_{4n+2p+5}, b$ are sent to the environment and evolved to `no`. The computation of the system halts, the answer of the system being *negative*.

### *3.2. Formal verification of the solution*

Next, we prove that the family $\mathbf{\Pi}$ of recognizer cell-like P systems from $\mathcal{CCEC}(2,2)$, designed in the previous subsection, provides a polynomial time and uniform solution to the `SAT` problem, according with Definition 3.

**Theorem 3.1.** $\texttt{SAT} \in \mathbf{PMC}_{\mathcal{CCEC}(2,2)}$.

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

- Every system of the family $\mathbf{\Pi}$ is a recognizer P systems from $\mathcal{CCEC}(2,2)$.
- The family $\mathbf{\Pi}$ is polynomially uniform by Turing machines because for each $n, p \in \mathbb{N}$, the rules of $\Pi(\langle n, p\rangle)$ of the family are recursively defined from $n, p \in \mathbb{N}$, and the amount of resources needed to build an element of the family is of a polynomial order in $n$ and $p$, as shown below:
  - Size of the working alphabet: $6np + 10n + 10p + 20 \in \Theta(np)$.
  - Initial number of membranes: $2 \in \Theta(1)$.
  - Initial number of objects: $3 \in \Theta(1)$.
  - Number of rules: $8n^2p + 4np + 12n - 2p + 9 \in \Theta(n^2p)$.
  - Maximal number of objects involved in any rule: $4 \in \Theta(1)$.
- The pair $(cod, s)$ of polynomial-time computable functions defined fulfill the following: for each instance $\varphi$ 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.
- The family $\mathbf{\Pi}$ is polynomially bounded: indeed, according to subsection 3.1, for each instance $\varphi$ of the `SAT` problem, the computation of $\Pi(s(\varphi)) + cod(\varphi)$ always halts and sends to the environment object `yes` (it takes at most $4n+2p+5$ steps) or object `no` (at step $4n+2p+6$), being $n$ the number of variables of $\varphi$ and $p$ the number of clauses.
- The family $\mathbf{\Pi}$ is sound with regard to $(X, cod, s)$: indeed, for each instance $\varphi$ of the `SAT` problem, if the computation of $\Pi(s(\varphi)) + cod(\varphi)$ is an accepting computation, then $\varphi$ is satisfiable.
- The family $\mathbf{\Pi}$ is complete with regard to $(X, cod, s)$: indeed, for each instance $\varphi$ of the `SAT` problem such that it is satisfiable, the computation of $\Pi(s(\varphi)) + cod(\varphi)$ is an accepting computation. □

**Corollary 3.1.** $\mathbf{NP} \cup \mathbf{co} - \mathbf{NP} \subseteq \mathbf{PMC}_{\mathcal{CCEC}(2,2)}$.

**Proof.** It suffices to make the following observations: the `SAT` problem is **NP**-complete, $\texttt{SAT} \in \mathbf{PMC}_{\mathcal{CCEC}(2,2)}$ and the class $\mathbf{PMC}_{\mathcal{CCEC}(2,2)}$ is closed under polynomial time reduction, and is under complementary. □

## 4. Conclusions and further works

In this work, a variant of cell-like P systems with evolutional communication rules, called cell-like P systems with evolutional communication rules and membrane creation has been proposed, and the computational efficiency of such P systems has been studied. In section 3, the presumed efficiency of $\mathcal{CCEC}(2,2)$ has been established by providing a polynomial time and uniform solution to the SAT problem by means of a family of recognizer membrane systems $\mathcal{CCEC}(2,2)$; that is by using recognizer cell-like P systems with evolutional communication rules and membrane creation such that the total number of objects involved in the LHS of the evolutional communication rules is at most 2 and the total number of objects involved in the RHS of the evolutional communication rule is at most 2.

The SAT problem has been solved by cell-like P systems from $\mathcal{CCEC}(2,2)$ in Section 3. It seems interesting to study the computational power of cell-like P systems from $\mathcal{CCEC}(2,1)$, $\mathcal{CCEC}(1,2)$ or $\mathcal{CCEC}(1,1)$.

In [24,41], a new strategy of using rules, called flat maximal parallelism was proposed, where in each membrane, a maximal set of applicable rules is chosen and each rule in the set is applied exactly once in each step. It would be interesting to investigate the presumed efficiency of cell-like P systems with evolutional communication rules and membrane creation by using rules in a flat maximally parallel way.

Time-free solutions to **NP**-complete problems have been studied widely [26,38–40,42], where the correctness of the solution is irrelevant to the times associated with the involved rules. It deserves to investigate the computational efficiency of cell-like P systems with evolutional communication rules and membrane creation by using rules in a time-free way.

The computational power of P systems with symport/antiport rules with respect to the number of objects and the number of membranes was investigated in [9,31]. An interesting open problem for P systems with evolutional symport/antiport rules is to find the minimal number of objects as well as the minimal number of membranes such that all recursively enumerable sets of natural numbers can be generated. Another interesting problem is to find characterizations of the sets of natural numbers for such kind of P systems that do not contain all recursively enumerable sets of natural numbers.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References


[1] A. Alhazov, R. Freund, P systems with one membrane and symport/antiport rules of five symbols are computationally complete, in: Proceedings of Third Brainstorming Week on Membrane Computing, Sevilla, Spain, 2005, pp. 19–28.
[2] A. Alhazov, C. Martín-Vide, L. Pan, Solving a PSPACE-complete problem by recognizing P systems with restricted active membranes, Fundam. Inform. 58 (2) (2003) 66–77.
[3] A. Alhazov, L. Pan, Gh. Păun, Trading polarizations for labels in P systems with active membranes, Acta Inform. 41 (2) (2004) 111–144.
[4] A. Alhazov, M.J. Pérez-Jiménez, Uniform solution of QSAT using polarizationless active membranes, Lect. Notes Comput. Sci. 4664 (2007) 122–133.
[5] A. Alhazov, Y. Rogozhin, Towards a characterization of P systems with minimal symport/antiport and two membranes, Lect. Notes Comput. Sci. 4361 (2006) 135–153.
[6] F. Bernardini, V. Manca, P systems with boundary rules, Lect. Notes Comput. Sci. 2597 (2003) 107–118.
[7] M. Cavaliere, Evolution-communication P systems, Lect. Notes Comput. Sci. 2597 (2003) 134–145.
[8] G. Ciobanu, L. Pan, Gh. Păun, M.J. Pérez-Jiménez, P systems with minimal parallelism, Theor. Comput. Sci. 378 (1) (2007) 117–130.
[9] E. Csuhaj-Varjú, M. Margenstern, G. Vaszil, S. Verlan, On small universal antiport P systems, Theor. Comput. Sci. 372 (2–3) (2007) 152–164.
[10] D. Díaz-Pernil, M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Núñez, A uniform family of tissue P system with cell division solving 3-COL in a linear time, Theor. Comput. Sci. 404 (1–2) (2008) 76–87.
[11] D. Díaz-Pernil, M.J. Pérez-Jiménez, A. Riscos-Núñez, Á. Romero-Jiménez, Computational efficiency of cellular division in tissue-like membrane systems, Rom. J. Inf. Sci. Technol. 11 (3) (2008) 229–241.
[12] P. Frisco, H.J. Hoogeboom, P systems with symport/antiport simulating counter automata, Acta Inform. 41 (2) (2004) 145–170.
[13] M.R. Garey, D.J. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, San Francisco, 1979.
[14] M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Núñez, F.J. Romero-Campero, Characterizing tractability with membrane creation, in: Proceedings of the Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC'05, 2005.
[15] M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, F.J. Romero-Campero, A linear solution of subset sum problem by using membrane creation, Lect. Notes Comput. Sci. 3561 (2005) 258–267.
[16] M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, F.J. Romero-Campero, A linear time solution for QSAT with membrane creation, Lect. Notes Comput. Sci. 3850 (2006) 241–252.
[17] M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, F.J. Romero-Campero, A uniform solution to SAT using membrane creation, Theor. Comput. Sci. 371 (1–2) (2007) 54–61.
[18] M. Ionescu, Gh. Păun, T. Yokomori, Spiking neural P systems, Fundam. Inform. 71 (2–3) (2006) 279–308.
[19] T.-O. Ishdorj, A. Leporati, L. Pan, X. Zeng, X. Zhang, Deterministic solutions to QSAT and Q3SAT by spiking neural P systems with pre-computed resources, Theor. Comput. Sci. 411 (25) (2010) 2345–2358.
[20] L.F. Macías-Ramos, M.J. Pérez-Jiménez, A. Riscos-Núñez, L. Valencia-Cabrera, Membrane fission versus cell division: when membrane proliferation is not enough, Theor. Comput. Sci. 608 (2015) 57–65.
[21] L.F. Macías-Ramos, B. Song, L. Valencia-Cabrera, L. Pan, M.J. Pérez-Jiménez, Membrane fission: a computational complexity perspective, Complexity 21 (6) (2016) 321–334.
[22] C. Martín-Vide, J. Pazos, Gh. Păun, A. Rodríguez-Patón, Tissue P systems, Theor. Comput. Sci. 296 (2) (2003) 295–326.
[23] M. Mutyam, K. Krithivasan, P systems with membrane creation: universality and efficiency, Lect. Notes Comput. Sci. 2055 (2001) 276–287.
[24] L. Pan, Gh. Păun, B. Song, Flat maximal parallelism in P systems with promoters, Theor. Comput. Sci. 623 (2016) 83–91.

[25] L. Pan, B. Song, L. Valencia-Cabrera, M.J. Pérez-Jiménez, The computational complexity of tissue P systems with evolutional symport/antiport rules, Complexity (2018) 3745210.
[26] L. Pan, X. Zeng, X. Zhang, Time-free spiking neural P systems, Neural Comput. 23 (2011) 1320–1342.
[27] A. Păun, Gh. Păun, The power of communication: P systems with symport/antiport, New Gener. Comput. 20 (3) (2002) 295–305.
[28] A. Păun, Gh. Păun, G. Rozenberg, Computing by communication in networks of membranes, Int. J. Found. Comput. 13 (6) (2002) 779–798.
[29] Gh. Păun, Computing with membranes, J. Comput. Syst. Sci. 61 (1) (2000) 108–143.
[30] Gh. Păun, Membrane Computing: An Introduction, Springer-Verlag, Berlin, 2002.
[31] Gh. Păun, J. Pazos, M.J. Pérez-Jiménez, A. Rodríguez-Patón, Symport/antiport P systems with three objects are universal, Fundam. Inform. 64 (1–4) (2005) 353–367.
[32] Gh. Păun, M.J. Pérez-Jiménez, A. Riscos-Núñez, Tissue P systems with cell division, Int. J. Comput. Commun. 3 (3) (2008) 295–303.
[33] Gh. Păun, G. Rozenberg, A. Salomaa (Eds.), The Oxford Handbook of Membrane Computing, Oxford University Press, New York, 2010.
[34] M.J. Pérez-Jiménez, Á. Romero-Jiménez, F. Sancho-Caparrini, Complexity classes in models of cellular computing with membranes, Nat. Comput. 2 (3) (2003) 265–285.
[35] M.J. Pérez-Jiménez, An approach to computational complexity in membrane computing, Lect. Notes Comput. Sci. 3365 (2005) 85–109.
[36] M.J. Pérez-Jiménez, P. Sosík, An optimal frontier of the efficiency of tissue P systems with cell separation, Fundam. Inform. 138 (1–2) (2015) 45–60.
[37] G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, 3 vols., Springer, Berlin, 1997.
[38] B. Song, Y. Kong, Solution to PSPACE-complete problem using P systems with active membranes with time-freeness, Math. Probl. Eng. (2019) 5793234.
[39] T. Song, L.F. Macías-Ramos, L. Pan, M.J. Pérez-Jiménez, Time-free solution to SAT problem using P systems with active membranes, Theor. Comput. Sci. 529 (2014) 61–68.
[40] B. Song, M.J. Pérez-Jiménez, L. Pan, An efficient time-free solution to QSAT problem using P systems with proteins on membranes, Inf. Comput. 256 (2017) 287–299.
[41] B. Song, M.J. Pérez-Jiménez, Gh. Păun, L. Pan, Tissue P systems with channel states working in the flat maximally parallel way, IEEE Trans. Nanobiosci. 15 (7) (2016) 645–656.
[42] B. Song, T. Song, L. Pan, A time-free uniform solution to subset sum problem by tissue P systems with cell division, Math. Struct. Comput. Sci. 27 (1) (2017) 17–32.
[43] B. Song, C. Zhang, L. Pan, Tissue-like P systems with evolutional symport/antiport rules, Inf. Sci. 378 (2017) 177–193.
[44] P. Sosík, P systems attacking hard problems beyond NP: a survey, J. Membr. Comput. 1 (3) (2019) 198–208.
[45] P. Sosík, A. Păun, A. Rodríguez-Patón, P systems with proteins on membranes characterize PSPACE, Theor. Comput. Sci. 488 (2013) 78–95.
[46] P. Sosík, A. Rodríguez-Patón, Membrane computing and complexity theory: a characterization of PSPACE, J. Comput. Syst. Sci. 73 (1) (2007) 137–152.