

---

# The Role of the Environment in Tissue P Systems with Cell Division

Mario J. Pérez-Jiménez<sup>1</sup>, Agustín Riscos-Núñez<sup>1</sup>, Miquel Rius-Font<sup>2</sup>,  
Francisco J. Romero-Campero<sup>1</sup>

<sup>1</sup> Research Group on Natural Computing  
Department of Computer Science and Artificial Intelligence  
University of Seville  
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain  
E-mail: [marper@us.es](mailto:marper@us.es), [ariscosn@us.es](mailto:ariscosn@us.es), [fran@us.es](mailto:fran@us.es)

<sup>2</sup> Department of Applied Mathematics IV  
Universitat Politècnica de Catalunya, Spain  
E-mail: [mr rius@ma4.upc.edu](mailto:mr rius@ma4.upc.edu)

**Summary.** Classical tissue P systems with cell division have a special alphabet whose elements appear at the initial configuration of the system in an arbitrary large number of copies. These objects are shared in a distinguished place of the system, called the environment. Besides, the ability of these computing devices to have infinite copies of some objects has been widely exploited in the design of efficient solutions to computationally hard problems.

This paper deals with computational aspects of tissue P systems with cell division where there is not an environment having the property mentioned above. Specifically, we establish the relationships between the polynomial complexity class associated with tissue P systems with cell division and with or without environment. As a consequence, we prove that it is not necessary to have infinite copies of some objects at the initial configuration in order to solve **NP**-complete problems in an efficient way.

**Key words:** Membrane Computing, Tissue P Systems, Cell Division, Environment of a tissue, Computational Complexity.

## 1 Preliminaries

An *alphabet*,  $\Gamma$ , is a non-empty set whose elements are called *symbols*. An ordered finite sequence of symbols is a *string* or *word*. If  $u$  and  $v$  are strings over  $\Gamma$ , then so is their *concatenation*  $uv$ , obtained by juxtaposition, that is, writing  $u$  and  $v$  one after the other. The number of symbols in a string  $u$  is the *length* of the string and it is denoted by  $|u|$ . As usual, the empty string (with length 0) will be denoted by  $\lambda$ . The set of all strings over an alphabet  $\Gamma$  is denoted by  $\Gamma^*$ . In algebraic terms,  $\Gamma^*$

is the free monoid generated by  $\Gamma$  under the operation of concatenation. Subsets, finite or infinite, of  $\Gamma^*$  are referred to as *languages* over  $\Gamma$ .

The set of symbols occurring in a string  $u \in \Gamma^*$  is denoted by  $\text{alph}(u)$ .

The *Parikh vector* associated with a string  $u \in \Gamma^*$  with respect to the alphabet  $\Sigma = \{a_1, \dots, a_r\} \subseteq \Gamma$  is  $\Psi_\Sigma(u) = (|u|_{a_1}, \dots, |u|_{a_r})$ , where  $|u|_{a_i}$  denotes the number of occurrences of symbol  $a_i$  in string  $u$ . This is called the *Parikh mapping* associated with  $\Sigma$ . Notice that, in this definition, the ordering of the symbols from  $\Sigma$  is relevant. If  $\Sigma_1 = \{a_{i_1}, \dots, a_{i_r}\} \subseteq \Gamma$ , then we define  $\Psi_{\Sigma_1}(u) = (|u|_{a_{i_1}}, \dots, |u|_{a_{i_r}})$ , for each  $u \in \Gamma^*$ .

A *multiset*  $m$  over a set  $A$  is a pair  $(A, f)$  where  $f : A \rightarrow \mathbb{N}$  is a mapping. If  $m = (A, f)$  is a multiset then its *support* is defined as  $\text{supp}(m) = \{x \in A \mid f(x) > 0\}$ . A multiset is empty (resp. finite) if its support is the empty set (resp. a finite set). If  $m = (A, f)$  is a finite multiset over  $A$  and  $\text{supp}(m) = \{a_1, \dots, a_k\}$ , then it will be denoted as  $m = \{a_1^{f(a_1)}, \dots, a_k^{f(a_k)}\}$ . That is, superscripts indicate the multiplicity of each element, and if  $f(x) = 0$  for  $x \in A$ , then element  $x$  is omitted. A finite multiset  $m = \{a_1^{f(a_1)}, \dots, a_k^{f(a_k)}\}$  can also be represented by the string  $a_1^{f(a_1)} \dots a_k^{f(a_k)}$  over the alphabet  $\{a_1, \dots, a_k\}$ . Nevertheless, all permutations of this string identify the same multiset  $m$  precisely. Throughout this paper, we speak about “the finite multiset  $m$ ” where  $m$  is a string, meaning “the finite multiset represented by the string  $m$ ”. If  $m_1 = (A, f_1)$ ,  $m_2 = (A, f_2)$  are multisets over  $A$ , then we define the union of  $m_1$  and  $m_2$  as  $m_1 + m_2 = (A, g)$ , where  $g = f_1 + f_2$ , that is,  $g(a) = f_1(a) + f_2(a)$ , for each  $a \in A$ .

For any sets  $A$  and  $B$  the *relative complement*  $A \setminus B$  of  $B$  in  $A$  is defined as follows:  $A \setminus B = \{x \in A \mid x \notin B\}$ .

Finally, for any set  $A$  we denote  $|A|$  the cardinal (number of elements) of  $A$ , as usual.

In what follows, we assume the reader is already familiar with the basic notions and terminology of P systems. For details, see [4].

## 2 Tissue P Systems with communication rules

**Definition 2.1** *A tissue P system with communication rules of degree  $q \geq 1$  is a tuple  $\Pi = (\Gamma, \mathcal{E}, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}, i_{out})$ , where:*

1.  $\Gamma$  is a finite alphabet whose elements are called objects;
2.  $\mathcal{E} \subseteq \Gamma$ ;
3.  $\mathcal{M}_1, \dots, \mathcal{M}_q$  are strings over  $\Gamma$ , representing finite multisets of objects;
4.  $\mathcal{R}$  is a finite set of communication rules of the form  $(i, u/v, j)$ , for  $i, j \in \{0, 1, 2, \dots, q\}$ ,  $i \neq j$ ,  $u, v \in \Gamma^*$ ,  $|u| + |v| > 0$ ;
5.  $i_{out} \in \{0, 1, 2, \dots, q\}$ .

A tissue P system *without environment* is a tissue P system such that  $\mathcal{E} = \emptyset$ . In this case, alphabet  $\mathcal{E}$  can be removed from the tuple.

A *tissue P system with communication rules*  $\Pi = (\Gamma, \mathcal{E}, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}, i_{out})$ , of degree  $q \geq 1$  can be viewed as a set of  $q$  cells, labelled by  $1, \dots, q$ , with an environment labelled by 0 such that: (a)  $\mathcal{M}_1, \dots, \mathcal{M}_q$  represent the finite multisets of objects initially placed in the  $q$  cells of the system; (b)  $\mathcal{E}$  is the set of objects initially located in the environment of the system, all of them available in an arbitrary number of copies; and (c)  $i_{out} \in \{0, 1, 2, \dots, q\}$  represents a distinguished cell or the environment which will encode the output of the system. We use the term *region*  $i$  ( $0 \leq i \leq q$ ) to refer cell  $i$  in the case  $1 \leq i \leq q$  and to refer the environment in the case  $i = 0$ .

When applying a rule  $(i, u/v, j)$ , the objects of the multiset represented by  $u$  are sent from region  $i$  to region  $j$  and, simultaneously, the objects of multiset  $v$  are sent from region  $j$  to region  $i$ . The length of the communication rule  $(i, u/v, j)$  is defined as  $|u| + |v|$ .

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

The rules of a system like the one above are used in a non-deterministic maximally parallel manner as it is customary in membrane computing. At each step, all cells which can evolve must evolve in a maximally parallel way (at each step we apply a multiset of rules which is maximal, no further applicable rule can be added).

An *instantaneous description* or a *configuration* at any instant of a tissue P system with communication rules is described by all multisets of objects over  $\Gamma$  associated with all the cells present in the system, and the multiset of objects over  $\Gamma - \mathcal{E}$  associated with the environment at that moment. Bearing in mind that the objects from  $\mathcal{E}$  have infinite copies in the environment, they are not properly changed along the computation. The *initial configuration* is  $(\mathcal{M}_1, \dots, \mathcal{M}_q; \emptyset)$ . A configuration is a *halting configuration* if no rule of the system is applicable to it.

Let us fix a tissue P system with communication rules  $\Pi$ . We say that configuration  $\mathcal{C}_1$  yields configuration  $\mathcal{C}_2$  in one *transition step*, denoted  $\mathcal{C}_1 \Rightarrow_{\Pi} \mathcal{C}_2$ , if we can pass from  $\mathcal{C}_1$  to  $\mathcal{C}_2$  by applying the rules from  $\mathcal{R}$  following the previous remarks. A *computation* of  $\Pi$  is a (finite or infinite) sequence of configurations such that:

1. the first term of the sequence is the initial configuration of the system;
2. each non-initial configuration of the sequence is obtained from the previous configuration by applying the rules of the system in a maximally parallel manner with the restrictions previously mentioned; and

3. if the sequence is finite (called *halting computation*), then the last term of the sequence is a halting configuration.

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

We denote by  $\mathbf{Comp}(\Pi)$  the set of computations of the tissue P system  $\Pi$ . If  $\mathcal{C} = \{\mathcal{C}_i\}_{i < r+1}$  of  $\Pi$  ( $r \in \mathbf{N}$ ) is a halting computation, then the *length of  $\mathcal{C}$*  is  $r$ , that is, the number of non-initial configurations which appear in the finite sequence  $\mathcal{C}$ . We denote it by  $|\mathcal{C}|$ . We also denote by  $\mathcal{C}_i(j)$  the contents of cell  $j$  at configuration  $\mathcal{C}_i$ .

### 3 Tissue P Systems with Cell Division

Cell division is an elegant process that enables organisms to grow and reproduce. Mitosis is a process of cell division which results in the production of two daughter cells from a single parent cell. Daughter cells are identical to one another and to the original parent cell. Through a sequence of steps, the replicated genetic material in a parent cell is equally distributed to two daughter cells. While there are some subtle differences, mitosis is remarkably similar across organisms.

Before a dividing cell enters mitosis, it undergoes a period of growth where the cell replicates its genetic material and organelles. Replication is one of the most important functions of a cell. DNA replication is a simple and precise process that creates two complete strands of DNA (one for each daughter cell) where only one existed before (from the parent cell).

Let us recall that the model of *tissue P systems with cell division* is based on the cell-like model of P systems with membranes division [3]. In these models, the cells are not polarized; the cells obtained by division have the same labels as the original cell, and if a cell is divided, its interaction with other cells or with the environment is locked during the division process. In some sense, this means that while a cell is dividing it closes its communication channels.

**Definition 3.1** *A tissue P system with cell division of degree  $q \geq 1$  is a tuple  $\Pi = (\Gamma, \mathcal{E}, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}, i_{out})$ , where:*

1.  $\Gamma$  is a finite alphabet whose elements are called objects;
2.  $\mathcal{E} \subseteq \Gamma$ ;
3.  $\mathcal{M}_1, \dots, \mathcal{M}_q$  are strings over  $\Gamma$ , representing finite multisets of objects;
4.  $\mathcal{R}$  is a finite set of rules of the following forms:
  - (a) Communication rules:  $(i, u/v, j)$ , for  $i, j \in \{0, 1, 2, \dots, q\}, i \neq j, u, v \in \Gamma^*, |u| + |v| > 0$ ;
  - (b) Division rules:  $[a]_i \rightarrow [b]_i[c]_i$ , where  $i \in \{1, 2, \dots, q\}, i \neq i_{out}$  and  $a, b, c \in \Gamma$ ;
5.  $i_{out} \in \{0, 1, 2, \dots, q\}$ .

A *tissue P system with cell division* is a tissue P system with communication rules where also division rules are allowed. When applying a division rule  $[a]_i \rightarrow [b]_i[c]_i$ , under the influence of object  $a$ , the cell with label  $i$  is divided into two cells with the same label; in the first copy, object  $a$  is replaced by object  $b$ , in the second one, object  $a$  is replaced by object  $c$ ; all the other objects residing in cell  $i$  are replicated and copies of them are placed in the two new cells. The output cell  $i_{out}$  cannot be divided.

The rules of a tissue P system with cell division are applied in a non-deterministic maximally parallel manner as it is customary in membrane computing. At each step, all cells which can evolve must evolve in a maximally parallel way (at each step we apply a multiset of rules which is maximal, no further applicable rule can be added), with the following important remark: if a cell divides, then the division rule is the only one which is applied for that cell at that step; the objects inside that cell do not evolve by means of communication rules. In other words, before division a cell interrupts all its communication channels with the other cells and with the environment. The new cells resulting from division will interact with other cells or with the environment only at the next step – providing that they do not divide once again. The label of a cell precisely identifies the rules which can be applied to it.

## 4 Recognizer Tissue P Systems

Let us recall that a *decision problem* is a pair  $(I_X, \theta_X)$  where  $I_X$  is a language over a finite alphabet (whose elements are called *instances*) and  $\theta_X$  is a total boolean function over  $I_X$ . Many abstract problems are not decision problems. For example, in *combinatorial optimization problems* some value must be optimized (minimized or maximized). In order to deal with such problems, they can be transformed into roughly equivalent decision problems by supplying a target/threshold value for the quantity to be optimized, and then asking whether this value can be attained.

A natural correspondence between decision problems and languages can be established as follows. Given a decision problem  $X = (I_X, \theta_X)$ , its associated language is  $L_X = \{w \in I_X : \theta_X(w) = 1\}$ . Conversely, given a language  $L$ , over an alphabet  $\Gamma$ , its associated decision problem is  $X_L = (I_{X_L}, \theta_{X_L})$ , where  $I_{X_L} = \Gamma^*$ , and  $\theta_{X_L} = \{(x, 1) : x \in L\} \cup \{(x, 0) : x \notin L\}$ . The solvability of decision problems is defined through the recognition of the languages associated with them.

In order to study the computing efficiency, the notions from classical *computational complexity theory* are adapted for membrane computing, and a special class of cell-like P systems is introduced in [7]: *recognizer P systems* (called *accepting P systems* in a previous paper [6]). For tissue P systems, with the same idea as recognizer cell-like P systems, *recognizer tissue P systems* is introduced in [5].

**Definition 4.1** A recognizer tissue P system with cell division of degree  $q \geq 1$  is a tuple  $\Pi = (\Gamma, \Sigma, \mathcal{E}, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}, i_{in}, i_{out})$ , where:

- $(\Gamma, \mathcal{E}, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}, i_{out})$  is a tissue P system with cell division of degree  $q \geq 1$ , as defined in the previous section.
- The working alphabet  $\Gamma$  has two distinguished objects **yes** and **no**, at least one copy of them present in some initial multisets  $\mathcal{M}_1, \dots, \mathcal{M}_q$ , but none of them is present in  $\mathcal{E}$ .
- $\Sigma$  is an (input) alphabet strictly contained in  $\Gamma$  such that  $\mathcal{E} \cap \Sigma = \emptyset$ .
- $\mathcal{M}_1, \dots, \mathcal{M}_q$  are strings over  $\Gamma \setminus \Sigma$ .
- $i_{in} \in \{1, \dots, q\}$  is the input cell.
- The output region  $i_{out}$  is the environment. In the case of tissue without environment,  $i_{out}$  is a distinguished cell, that is  $i_{out} \in \{1, \dots, q\}$ .
- All computations halt.
- If  $\mathcal{C}$  is a computation of  $\Pi$ , then either object **yes** or object **no** (but not both) must have been released into the environment, and only at the last step of the computation.

For each multiset  $m$  over  $\Sigma$ , the computation of the system  $\Pi$  with input  $m$  starts from the configuration of the form  $(\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_{i_{in}} + m, \dots, \mathcal{M}_q; \emptyset)$ , that is, the input multiset  $m$  has been added to the contents of the input cell  $i_{in}$ , and we denote it by  $\Pi + m$ . Therefore, we have an initial configuration associated with each input multiset  $m$  (over the input alphabet  $\Sigma$ ) in this kind of systems.

Given a recognizer tissue P system with cell division, and a halting computation  $\mathcal{C} = \{\mathcal{C}_i\}_{i < r+1}$  of  $\Pi$  ( $r \in \mathbf{N}$ ), we define the result of  $\mathcal{C}$  as follows:

$$Output(\mathcal{C}) = \begin{cases} \text{yes,} & \text{if } \Psi_{\{\text{yes, no}\}}(M_{r, i_{out}}) = (1, 0) \wedge \\ & \Psi_{\{\text{yes, no}\}}(M_{i, i_{out}}) = (0, 0) \text{ for } i = 0, \dots, r-1 \\ \text{no,} & \text{if } \Psi_{\{\text{yes, no}\}}(M_{r, i_{out}}) = (0, 1) \wedge \\ & \Psi_{\{\text{yes, no}\}}(M_{i, i_{out}}) = (0, 0) \text{ for } i = 0, \dots, r-1 \end{cases}$$

where  $\Psi$  is the Parikh mapping, and  $M_{i, i_{out}}$  is the multiset over  $\Gamma \setminus \mathcal{E}$  associated with the output region at the configuration  $\mathcal{C}_i$ , in particular,  $M_{r, i_{out}}$  is the multiset over  $\Gamma \setminus \mathcal{E}$  associated with the output region at the halting configuration  $\mathcal{C}_r$ .

We say that a computation  $\mathcal{C}$  is an *accepting computation* (respectively, *rejecting computation*) if  $Output(\mathcal{C}) = \text{yes}$  (respectively,  $Output(\mathcal{C}) = \text{no}$ ), that is, if object **yes** (respectively, object **no**) appears in the output region associated with the corresponding halting configuration of  $\mathcal{C}$ , and neither object **yes** nor **no** appears in the output region associated with any non-halting configuration of  $\mathcal{C}$ .

Let us notice that if a recognizer tissue P system

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

has a rule of the type  $(i, \lambda/u, 0)$  then  $alph(u) \cap (\Gamma \setminus \mathcal{E}) \neq \emptyset$ , because on the contrary all computations of  $\Pi$  would be non halting.

For each natural number  $k \geq 1$ , we denote by  $\mathbf{TDC}(k)$  the class of recognizer tissue P systems with cell division and with communication rules of length at most  $k$ . In the case of tissue P systems without environment, we denote by  $\widehat{\mathbf{TDC}}(k)$  the class of recognizer tissue P systems with cell division and with communication rules of length at most  $k$ .

## 5 Polynomial Complexity Classes of Tissue P systems

Next, we define what solving a decision problem in the framework of tissue P systems in a uniform and efficient way means. Bearing in mind that they provide devices with a finite description, a numerable family of tissue P systems will be necessary in order to solve a decision problem.

**Definition 5.1** *We say that a decision problem  $X = (I_X, \theta_X)$  is solvable in a uniform way and polynomial time by a family  $\Pi = \{\Pi(n) \mid n \in \mathbb{N}\}$  of recognizer tissue P systems (with symport/antiport rules, with cell division or with cell separation) if the following holds:*

- *The family  $\Pi$  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  $\Pi(s(u))$ ;*
  - *for each  $n \in \mathbb{N}$ ,  $s^{-1}(n)$  is a finite set;*
  - *the family  $\Pi$  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  $\Pi$  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  $\Pi$  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.*

From the soundness and completeness conditions above we deduce that every P system  $\Pi(n)$  is *confluent*, in the following sense: every computation of a system with the *same* input multiset must always give the *same* answer.

Let  $\mathbf{R}$  be a class of recognizer tissue P systems. We denote by  $\mathbf{PMC}_{\mathbf{R}}$  the set of all decision problems which can be solved in a uniform way and polynomial time by means of families of systems from  $\mathbf{R}$ . The class  $\mathbf{PMC}_{\mathbf{R}}$  is closed under complement and polynomial-time reductions [6].

Next, we prove a technical result concerning recognizer tissue P systems.

**Lemma 5.2** *Let  $\Pi = \{\Pi(n) \mid n \in \mathbb{N}\}$  a family of recognizer tissue P systems solving a decision problem  $X = (I_X, \theta_X)$  in polynomial time according to the previous definition. Let  $(cod, s)$  a polynomial encoding associated with that solution. Let  $r(n)$  be a polynomial function such that for each  $u \in I_X$  every computation of  $\Pi(s(u)) + cod(u)$  is halting and it performs at most  $r(|u|)$  steps. Then, there exists a polynomial function  $p(n)$  such that for each instance  $u \in I_X$ ,  $2^{p(|u|)}$  is an*

upper bound of the number of objects from  $\mathcal{E}$  which are moved from the environment to all cells of the system  $\Pi(s(u)) + \text{cod}(u)$  by communication rules along any computation.

**Proof:** Let  $u \in I_X$  be an instance of  $X$  and

$$\Pi(s(u)) + \text{cod}(u) = (\Gamma, \Sigma, \mathcal{E}, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}, i_{in}, i_{out})$$

Let  $k \in \mathbb{N}$  be such that  $\Pi(s(u)) + \text{cod}(u) \in \mathbf{TDC}(k)$ . Let  $M = |\mathcal{M}_1 + \dots + \mathcal{M}_q|$ . Then, any computation of  $\Pi(s(u)) + \text{cod}(u)$  performs, at most,  $r(|u|)$  transition steps. Let  $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_m)$ ,  $0 \leq m \leq r(|u|)$ , be a computation of  $\Pi$ . For each  $t$ ,  $0 \leq t \leq m$  and  $i$ ,  $1 \leq i \leq q$ , we denote by  $\mathcal{C}_t(i)$  the multiset of objects over  $\Gamma$  in cell  $i$  at time  $t$ . We also denote  $\mathcal{C}_t(0)$  the multiset of objects over  $\Gamma \setminus \mathcal{E}$  in the environment at time  $t$ .

Let us suppose that we apply only communication rules at  $m$  consecutive transition steps. At this situation, for each  $t$  ( $0 \leq t \leq m$ ) we compute an upper bound of  $|\mathcal{C}_t(0) + \mathcal{C}_t(1) + \dots + \mathcal{C}_t(q)|$ . Then, for each  $i, j$  ( $0 \leq i, j \leq q$ ,  $i \neq j$ ) we denote by  $A_t(i, j)$  the multiset of objects being moved from region  $j$  to region  $i$  by applying rules of the type  $(i, u/v, j)$  at time  $t$ .

Let us construct  $\alpha_t$ ,  $0 \leq t \leq m$ , an upper bound of the number of objects which appear in the whole system (taking all cells into account) at time  $t$ . That is,

$$\alpha_t \geq |\mathcal{C}_t(0) + \mathcal{C}_t(1) + \dots + \mathcal{C}_t(q)|$$

The construction is made by induction on  $t$ . For  $t = 0$  we consider  $\alpha_0 = M$ . Let  $t$  be such that  $0 \leq t < m$  and for each  $t'$  ( $0 \leq t' \leq t$ ) let us assume that we have constructed  $\alpha_{t'}$  such that

$$\alpha_{t'} \geq \sum_{i=0}^q |\mathcal{C}_{t'}(i)|$$

The number of objects moved into cell  $i$  ( $1 \leq i \leq q$ ) at instant  $t + 1$  is

$$A_t(i, 0) + \sum_{j=1, j \neq i}^q A_t(i, j)$$

The number of objects sent to the environment at instant  $t + 1$  is  $\sum_{j=1}^q A_t(0, j)$ .

Notice that objects coming to region  $i$  from some other cell  $j$  were already present in the previous configuration. Besides, in order to trigger a communication rule bringing objects from the environment into region  $i$ , at least one object in region  $i$  is required, or else one symbol from  $\Gamma \setminus \mathcal{E}$  in the environment. Finally, recall that the length of communication rules is bounded by  $k$ .

From these considerations, we deduce:

$$\sum_{i=1}^q \sum_{j=1, j \neq i}^q |A_t(i, j)| \leq \alpha_t \quad \text{and} \quad \sum_{i=1}^q |A_t(i, 0)| \leq \alpha_t \cdot k$$



Besides,

$$\sum_{j=1}^q |A_t(0, j)| \leq \alpha_t \cdot k$$

Then, we can consider  $\alpha_{t+1} = \alpha_t + \alpha_t \cdot k + \alpha_t \cdot k = \alpha_t \cdot (1 + 2k)$ . Thus, for each  $t$  ( $0 \leq t \leq m$ ) we define  $\alpha_t = M \cdot (1 + 2k)^t$ . Hence, if we applied in a consecutive way the maximum possible number of communication rules (without applying any division rules) to the system  $\Pi(s(u)) + cod(u)$ , in any instant of any computation of the system,  $M \cdot (1 + 2k)^{r(|u|)}$  is an upper bound of the number of objects in the whole system.

Now, let us consider the effects of applying in a consecutive way the maximum possible number of division rules (without applying any communication rules) to the system  $\Pi(s(u)) + cod(u)$  when the initial configuration has  $M \cdot (1 + 2k)^{r(|u|)}$  objects. After that, an upper bound of the number of objects in the whole system by any computation is  $M \cdot (1 + 2k)^{r(|u|)} \cdot 2^{r(|u|)} \cdot r(|u|)$ . Hence, for each instance  $u \in I_X$  the number of objects from  $\mathcal{E}$  which are moved from the environment to the whole cells of the system  $\Pi(s(u)) + cod(u)$  is, at most,  $M \cdot (1 + 2k)^{r(|u|)} \cdot 2^{r(|u|)} \cdot r(|u|)$ .

Then, we consider a polynomial function  $p(n)$  such that

$$p(|u|) \geq \log(M) + r(|u|) \cdot \log(1 + 2k) + r(|u|) + \log(r(|u|))$$

for each instance  $u \in I_X$ . The polynomial function  $p(n)$  fulfills the property required at the Lemma. □

## 6 Simulating tissue P systems with cell division by means of tissue P systems with cell division and without environment

The goal of this section is to show that any tissue P system with cell division can be simulated by a tissue P system with cell division and without environment in an efficient way.

First of all, we define the meaning of efficient simulations in the framework of recognizer tissue P systems.

**Definition 6.1** *Let  $\Pi$  and  $\Pi'$  be recognizer tissue P systems. We say that  $\Pi'$  simulates  $\Pi$  in an efficient way if the following holds:*

1.  $\Pi'$  can be constructed from  $\Pi$  by a deterministic Turing machine working in polynomial time.
2. There exists an injective function,  $f$ , from the set  $\mathbf{Comp}(\Pi)$  of computations of  $\Pi$  onto the set  $\mathbf{Comp}(\Pi')$  of computations of  $\Pi'$  such that:
  - ★ There exists a deterministic Turing machine that constructs computation  $f(\mathcal{C})$  from computation  $\mathcal{C}$  in polynomial time.
  - ★ A computation  $\mathcal{C} \in \mathbf{Comp}(\Pi)$  is an accepting computation if and only if  $f(\mathcal{C}) \in \mathbf{Comp}(\Pi')$  is an accepting one.

- ★ *There exists a polynomial function  $p(n)$  such that for each  $C \in \mathbf{Comp}(\Pi)$  we have  $|f(C)| \leq p(|C|)$ .*

Now, for every family of recognizer tissue P system with cell division solving a decision problem, we design a family of recognizer tissue P systems with cell division and *without environment* efficiently simulating it, according to Definition 6.1.

In what follows throughout this Section, let  $\mathbf{\Pi} = \{\Pi(n) \mid n \in \mathbb{N}\}$  a family of recognizer tissue P systems solving a decision problem  $X = (I_X, \theta_X)$  in polynomial time according to Definition 5.1, and let  $p(n)$  be a polynomial function such that for each instance  $u \in I_X$ ,  $2^{p(|u|)}$  is an upper bound of the number of objects from  $\mathcal{E}$  which are moved from the environment to all cells of the system by any computation of  $\Pi(s(u)) + \text{cod}(u)$ .

**Definition 6.2** *For each  $n \in \mathbb{N}$ , let  $\Pi(n) = (\Gamma, \Sigma, \mathcal{E}, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}, i_{in}, i_{out})$  an element of the previous family of degree  $q$  and for the sake of simplicity we denote  $p$  instead of  $p(n)$ . Let us consider the recognizer tissue P system of degree  $q_1 = 1 + q \cdot (p + 2) + |\mathcal{E}|$  with cell division and without environment*

$$\mathbf{S}(\Pi(n)) = (\Gamma', \Sigma', \mathcal{M}'_0, \mathcal{M}'_1, \dots, \mathcal{M}'_{q_1}, \mathcal{R}', i'_{in}, i'_{out})$$

defined as follows:

- $\Gamma' = \Gamma \cup \{\alpha_i : 0 \leq i \leq p - 1\}$ .
- $\Sigma' = \Sigma$ .
- Each cell  $i \in \{1, \dots, q\}$  of  $\Pi$  provides a cell of  $\mathbf{S}(\Pi(n))$  with the same label. In addition,  $\mathbf{S}(\Pi(n))$  has:
  - $p + 1$  new cells, labelled by  $(i, 0), (i, 1), \dots, (i, p)$ , respectively, for each  $i \in \{1, \dots, q\}$ .
  - A distinguished cell labelled by 0.
  - A new cell, labelled by  $l_b$ , for each  $b \in \mathcal{E}$ .
- Initial multisets:  $\mathcal{M}'_{l_b} = \{\alpha_0\}$ , for each  $b \in \mathcal{E}$ , and

$$\left. \begin{array}{l} \mathcal{M}'_{(i,0)} = \mathcal{M}_i \\ \mathcal{M}'_{(i,1)} = \emptyset \\ \dots\dots\dots \dots \\ \mathcal{M}'_{(i,p)} = \emptyset \\ \mathcal{M}'_i = \emptyset \end{array} \right\} (1 \leq i \leq q)$$

- Set of rules:

$$\begin{aligned} \mathcal{R}' = & \mathcal{R} \cup \{[\alpha_j]_{l_b} \rightarrow [\alpha_{j+1}]_{l_b} \mid [\alpha_{j+1}]_{l_b} : b \in \mathcal{E} \wedge 0 \leq j \leq p - 2\} \\ & \cup \{[\alpha_{p-1}]_{l_b} \rightarrow [b]_{l_b} \mid [b]_{l_b} : b \in \mathcal{E}\} \\ & \cup \{(l_b, b/\lambda, 0) : b \in \mathcal{E}\} \\ & \cup \{((i, j), a/\lambda, (i, j + 1)) : a \in \Gamma \wedge 1 \leq i \leq q \wedge 0 \leq j \leq p - 1\} \\ & \cup \{((i, p), a/\lambda, i) : a \in \Gamma \wedge 1 \leq i \leq q\} \end{aligned}$$

- $i'_{in} = (i_{in}, 0)$ , and  $i'_{out} = 0$ .

Let us notice that  $\mathbf{S}(\Pi(n))$  can be considered as an **extension** of  $\Pi(n)$  **without environment**, in the following sense:

- ★  $\Gamma \subseteq \Gamma', \Sigma \subseteq \Sigma'$  and  $\mathcal{E} = \emptyset$ .
- ★ Each cell in  $\Pi$  is also a cell in  $\mathbf{S}(\Pi(n))$ .
- ★ There is a distinguished cell in  $\mathbf{S}(\Pi(n))$  labelled by 0 which plays the role of environment of  $\Pi(n)$ .
- ★  $\mathcal{R} \subseteq \mathcal{R}'$ , and now 0 is the label of a “normal cell” in  $\mathbf{S}(\Pi(n))$ .

Next, we analyze the structure of the computations of system  $\mathbf{S}(\Pi(n))$  and we compare them with the computations of  $\Pi(n)$ .

**Lemma 6.3** *Let  $\mathcal{C}' = (\mathcal{C}'_0, \mathcal{C}'_1, \dots)$  be a computation of  $\mathbf{S}(\Pi(n))$ . For each  $t$  ( $1 \leq t \leq p$ ) the following holds:*

- $\mathcal{C}'_t(i) = \emptyset$ , for  $0 \leq i \leq q$ .
- For each  $1 \leq i \leq q$ , and  $0 \leq j \leq p$  we have:

$$\mathcal{C}'_t(i, j) = \begin{cases} \mathcal{M}_i, & \text{if } j = t \\ \emptyset, & \text{if } j \neq t \end{cases}$$

- For each  $b \in \mathcal{E}$ , there exist  $2^t$  cells labelled by  $l_b$  whose content is:

$$\mathcal{C}'_t(l_b) = \begin{cases} \alpha_t, & \text{if } 1 \leq t \leq p-1 \\ b, & \text{if } t = p \end{cases}$$

**Proof:** By induction on  $t$ .

Let us start with the basic case  $t = 1$ . The initial configuration of system  $\mathbf{S}(\Pi(n))$  is the following:

- $\mathcal{C}'_0(i) = \emptyset$ , for  $0 \leq i \leq q$ .
- For each  $1 \leq i \leq q$  we have  $\mathcal{C}'_0(i, 0) = \mathcal{M}_i$ , and  $\mathcal{C}'_0(i, j) = \emptyset$ , for  $1 \leq j \leq p$ .
- For each  $b \in \mathcal{E}$ , there exists only one cell labelled by  $l_b$  whose contents is  $\{\alpha_0\}$ .

At configuration  $\mathcal{C}'_0$ , only the following rules are applicable:

- $[\alpha_0]_{l_b} \rightarrow [\alpha_1]_{l_b} [\alpha_1]_{l_b}$ , for each  $b \in \mathcal{E}$ .
- $((i, 0), a/\lambda, (i, 1))$ , for each  $a \in \text{supp}(\mathcal{M}_i)$ .

Thus,

- For each  $i$  ( $1 \leq i \leq q$ ) we have:

$$\begin{cases} \mathcal{C}'_1(i) & = \emptyset \\ \mathcal{C}'_1(0) & = \emptyset \\ \mathcal{C}'_1(i, 0) & = \emptyset \\ \mathcal{C}'_1(i, 1) & = \mathcal{M}_i \\ \mathcal{C}'_1(i, j) & = \emptyset, \text{ for } 2 \leq j \leq p \end{cases}$$

- For each  $b \in \mathcal{E}$ , there are 2 cells labelled by  $l_b$  whose content is  $\{\alpha_1\}$ .

Hence, the result holds for  $t = 1$ .

By induction hypothesis, let  $t$  be such that  $1 \leq t < p$ , and let us suppose the result holds for  $t$ , that is,

- $\mathcal{C}'_t(i) = \emptyset$ , for  $0 \leq i \leq q$ .
- For each  $1 \leq i \leq q$ , and  $0 \leq j \leq p$  we have:

$$\mathcal{C}'_t(i, j) = \begin{cases} \mathcal{M}_i, & \text{if } j = t \\ \emptyset, & \text{if } j \neq t \end{cases}$$

- For each  $b \in \mathcal{E}$ , there exist  $2^t$  cells labelled by  $l_b$  whose contents is  $\mathcal{C}'_t(l_b) = \{\alpha_t\}$  (because  $t \leq p - 1$ ).

Then, at configuration  $\mathcal{C}'_t$  only the following rules are applicable:

- (1) If  $t \leq p - 2$ , the rules  $[\alpha_t]_{l_b} \rightarrow [\alpha_{t+1}]_{l_b} [\alpha_{t+1}]_{l_b}$ , for each  $b \in \mathcal{E}$ .
- (2) If  $t = p - 1$ , the rules  $[\alpha_{p-1}]_{l_b} \rightarrow [b]_{l_b} [b]_{l_b}$ , for each  $b \in \mathcal{E}$ .
- (3)  $((i, t), a/\lambda, (i, t + 1))$ , for each  $a \in \Gamma$ .

From the application of rules of types (1) or (2) at configuration  $\mathcal{C}'_t$  we deduce that there are  $2^{t+1}$  cells labelled by  $l_b$  in  $\mathcal{C}'_{t+1}$ , for each  $b \in \mathcal{E}$ , whose content is  $\{\alpha_{t+1}\}$ , if  $t \leq p - 2$ , or  $\{b\}$ , if  $t = p - 1$ .

From the application of rules of type (3) at configuration  $\mathcal{C}'_t$ , we deduce that

$$\mathcal{C}'_{t+1}(i, j) = \begin{cases} \mathcal{M}_i, & \text{if } j = t + 1 \\ \emptyset, & \text{if } 0 \leq j \leq p \wedge j \neq t + 1 \end{cases}$$

Bearing in mind that no other rule of system  $\mathbf{S}(\Pi(n))$  is applicable, we deduce that  $\mathcal{C}'_{t+1}(i) = \emptyset$ , for  $0 \leq i \leq q$ .

This completes the proof of this Lemma. □

**Lemma 6.4** *Let  $\mathcal{C}' = (\mathcal{C}'_0, \mathcal{C}'_1, \dots)$  be a computation of the tissue  $P$  system  $\mathbf{S}(\Pi(n))$ . Configuration  $\mathcal{C}'_{p+1}$  is the following:*

- (1)  $\mathcal{C}'_{p+1}(0) = b_1^{2^p} \dots b_m^{2^p}$ , where  $\mathcal{E} = \{b_1, \dots, b_m\}$ .
- (2)  $\mathcal{C}'_{p+1}(i) = \mathcal{M}_i = C_0(i)$ , for  $1 \leq i \leq q$ .
- (3)  $\mathcal{C}'_{p+1}(i, j) = \emptyset$ , for  $1 \leq i \leq q$ ,  $0 \leq j \leq p$ .
- (4) *There exist  $2^p$  cells labelled by  $l_b$  whose content is empty, for  $b \in \mathcal{E}$ .*

**Proof:** From Lemma 6.3, the configuration  $\mathcal{C}'_p$  is the following:

- $\mathcal{C}'_p(i) = \emptyset$ , for  $0 \leq i \leq q$ .
- For each  $i$  ( $1 \leq i \leq q$ ) we have

$$\mathcal{C}'_p(i, j) = \begin{cases} \mathcal{M}_i, & \text{if } j = p \\ \emptyset, & \text{if } j \neq p \end{cases}$$

- For each  $b \in \mathcal{E}$ , there exist  $2^p$  cells labelled by  $l_b$  whose content is  $\{b\}$ .

At configuration  $\mathcal{C}'_p$  only the following rules are applicables:

- $((i, p), a/\lambda, i)$ , for each  $a \in \Gamma \cap \text{supp}(\mathcal{M}_i)$ .
- $(l_b, b/\lambda, 0)$ , for each  $b \in \mathcal{E}$ .

Thus,

- $\mathcal{C}'_{p+1}(0) = b_1^{2^p} \dots b_m^{2^p}$ , where  $\mathcal{E} = \{b_1, \dots, b_m\}$ .
- $\mathcal{C}'_{p+1}(i) = \mathcal{M}_i = C_0(i)$ , for  $1 \leq i \leq q$ .
- $\mathcal{C}'_{p+1}(i, j) = \emptyset$ , for  $1 \leq i \leq q$  and  $0 \leq j \leq p$ .
- There exist  $2^p$  cells labelled by  $l_b$  whose content is empty, for each  $b \in \mathcal{E}$ .

□

**Definition 6.5** Let  $\mathcal{C} = (C_0, C_1, \dots, C_r)$  be a halting computation of  $\Pi(n)$ . Then we define the computation  $\mathbf{S}(\mathcal{C}) = (C'_0, C'_1, \dots, C'_p, C'_{p+1}, \dots, C'_{p+1+r})$  of  $\mathbf{S}(\Pi(n))$  as follows:

- (1) The initial configuration is:
 
$$\begin{cases} \mathcal{C}'_0(i) = \emptyset, & \text{for } 0 \leq i \leq q \\ \mathcal{C}'_0(i, 0) = C_0(i), & \text{for } 1 \leq i \leq q \\ \mathcal{C}'_0(i, j) = \emptyset, & \text{for } 1 \leq i \leq q \text{ and } 1 \leq j \leq p \\ \mathcal{C}'_0(l_b) = \alpha_0, & \text{for each } b \in \mathcal{E} \end{cases}$$
- (2) The configuration  $\mathcal{C}'_t$ , for  $1 \leq t \leq p$ , is described by Lemma 6.3.
- (3) The configuration  $\mathcal{C}'_{p+1}$  is described by Lemma 6.4.
- (4) The configuration  $\mathcal{C}'_{p+1+s}$ , for  $0 \leq s \leq r$ , coincides with the configuration  $C_s$  of  $\Pi$ , that is,  $\mathcal{C}'_{p+1+s}(i) = C_s(i)$ , for  $1 \leq i \leq q$ . The content of the remaining cells (excluding cell 0) at configuration  $\mathcal{C}'_{p+1+s}$  is equal to the content of that cell at configuration  $\mathcal{C}'_{p+1}$ , that is, these cells do not evolve after step  $p+1$ .

That is, every computation  $\mathcal{C}$  of  $\Pi(n)$  can be “reproduced” by a computation  $\mathbf{S}(\mathcal{C})$  of  $\mathbf{S}(\Pi(n))$  with a delay: from step  $p+1$  to step  $p+1+r$  the computation  $\mathbf{S}(\mathcal{C})$  restricted to cells  $1, \dots, q$  provides the computation  $\mathcal{C}$  of  $\Pi(n)$ .

From Lemma 6.3 and Lemma 6.4 we deduce that: (a)  $\mathbf{S}(\mathcal{C})$  is a computation of  $\mathbf{S}(\Pi(n))$ , and (b)  $S$  is an injective function from  $\mathbf{Comp}(\Pi(n))$  onto  $\mathbf{Comp}(\mathbf{S}(\Pi(n)))$ . Moreover, if  $p$  is a polynomial function on the size of  $\Pi(n)$ , then we have the following:

**Proposition 6.6** The tissue P system  $\mathbf{S}(\Pi(n))$  defined in 6.2 simulates  $\Pi(n)$  in an efficient way.

*Proof.* In order to show that  $\mathbf{S}(\Pi(n))$  can be constructed from  $\Pi(n)$  by a deterministic Turing machine working in polynomial time, it is enough to note that the amount of resources needed to construct  $\mathbf{S}(\Pi(n))$  from  $\Pi(n)$  is polynomial in the size of the initial resources of  $\Pi(n)$ . Indeed,

1. The size of the alphabet of  $\mathbf{S}(\Pi(n))$  is  $|\Gamma'| = |\Gamma| + p$ .

2. The initial number of cells of  $\mathbf{S}(\Pi(n))$  is  $1 + q \cdot (p + 2) + |\mathcal{E}|$ .
3. The initial number of objects of  $\mathbf{S}(\Pi(n))$  is the initial number of objects of  $\Pi(n)$  plus  $|\mathcal{E}|$ .
4. The number of rules of  $\mathbf{S}(\Pi(n))$  is  $|\mathcal{R}'| = |\mathcal{R}| + (p + 1) \cdot |\mathcal{E}| + |\Gamma| \cdot q \cdot (p + 1)$ .
5. The maximal length of a communication rule of  $\mathbf{S}(\Pi(n))$  is equal to the maximal length of a communication rule of  $\Pi(n)$ .

From Lemma 6.3 and Lemma 6.4 we deduce that: (a) every computation  $\mathcal{C}'$  of  $\mathbf{S}(\Pi(n))$  has associated a computation  $\mathcal{C}$  of  $\Pi(n)$  such that  $\mathbf{S}(\mathcal{C}) = \mathcal{C}'$  in a natural way, (b) the function  $S$  is injective, and (c) a computation  $\mathcal{C}$  of  $\Pi$  is an accepting computation if and only if  $\mathbf{S}(\mathcal{C})$  is an accepting computation of  $\mathbf{S}(\Pi(n))$ .

Finally, let us notice that if  $\mathcal{C}$  is a computation of  $\Pi(n)$  with length  $r$ , then  $\mathbf{S}(\mathcal{C})$  is a computation of  $\mathbf{S}(\Pi(n))$  with length  $p + 1 + r$ .

## 7 Computational Complexity classes of Tissue P Systems with Cell Division and without environment

In this Section, we analyze the role of the environment in the efficiency of tissue P systems with cell division. That is, we study the ability of these P systems with respect to the computational efficiency when the alphabet of the environment is an empty set.

**Theorem 7.1** *For each  $k \in \mathbb{N}$  we have  $\mathbf{PMC}_{\mathbf{TDC}(k+1)} = \mathbf{PMC}_{\widehat{\mathbf{TDC}}(k+1)}$ .*

**Proof:** Obviously,  $\mathbf{P} \subseteq \mathbf{PMC}_{\widehat{\mathbf{TDC}}(1)} \subseteq \mathbf{PMC}_{\mathbf{TDC}(1)} = \mathbf{P}$ .

Let  $k \geq 1$ . Since  $\widehat{\mathbf{TDC}}(k+1) \subseteq \mathbf{TDC}(k+1)$  it suffices to show that  $\mathbf{PMC}_{\mathbf{TDC}(k+1)} \subseteq \mathbf{PMC}_{\widehat{\mathbf{TDC}}(k+1)}$ . For that, let  $X \in \mathbf{PMC}_{\mathbf{TDC}(k+1)}$ . Let us show that  $X \in \mathbf{PMC}_{\widehat{\mathbf{TDC}}(k+1)}$ .

Let  $\{\Pi(n) : n \in N\}$  be a family of tissue P systems from  $\mathbf{TDC}(k+1)$  solving  $X$  according to Definition 5.1. Let  $(cod, s)$  be a polynomial encoding associated with that solution. Let  $u \in I_X$  be an instance of the problem  $X$  and  $s(u) = n$ . Then, that instance will be processed by the system  $\Pi(s(u)) + cod(u)$ . According to Lemma 5.2, let  $p(n)$  be a polynomial function such that  $2^{p(|u|)}$  is an upper bound of the number of objects from  $\mathcal{E}$  which are moved from the environment to all cells of the system by any computation of  $\Pi(s(u)) + cod(u)$ , for each instance  $u \in I_X$ .

If  $\Pi(s(u)) + cod(u) = (\Gamma, \Sigma, \mathcal{M}_1, \dots, \mathcal{M}_{i_{in}} + cod(u), \dots, \mathcal{M}_q, \mathcal{E}, \mathcal{R}, i_{in}, i_{out})$ , we consider the tissue P system without environment

$$\mathbf{S}(\Pi(s(u))) + cod(u) = (\Gamma', \Sigma', \mathcal{M}'_0, \mathcal{M}'_1, \dots, \mathcal{M}'_{i_{in}} + cod(u), \dots, \mathcal{M}'_{q_1}, \mathcal{R}', i'_{in}, i'_{out})$$

according to Definition 6.2, where  $q_1 = 1 + q \cdot (p(|u|) + 2) + |\mathcal{E}|$ .

Therefore,  $\mathbf{S}(\Pi(s(u))) + cod(u) \in \widehat{\mathbf{TDC}}(k+1)$  and in the system  $\mathbf{S}(\Pi(s(u))) + cod(u)$  the following holds:

- A new distinguished cell labelled by 0 has been considered, which will play the role of the environment at the system  $\Pi(s(u)) + cod(u)$ .
- We must guarantee that system  $\mathbf{S}(\Pi(s(u))) + cod(u)$  has initially enough objects in cell 0 to simulate the behaviour of the environment of  $\Pi(n)$ .
- New objects, new rules and new cells will be introduced in  $\mathbf{S}(\Pi(s(u))) + cod(u)$ .
- After  $p(n) + 1$  step, computations of  $\mathbf{S}(\Pi(s(u))) + cod(u)$  reproduce the computations of  $\Pi(s(u)) + cod(u)$  exactly.

Let us suppose that  $\mathcal{E} = \{b_1, \dots, b_m\}$ . In order to simulate  $\Pi(s(u)) + cod(u)$  by a tissue P system without environment in an efficient way, we need to have enough objects in the cell of  $\mathbf{S}(\Pi(s(u))) + cod(u)$  labelled by 0 available. That is,  $2^{p(n)}$  objects in that cell are enough.

In order to start the simulation of any computation  $\mathcal{C}$  of  $\Pi(s(u)) + cod(u)$ , it would be enough to have  $2^{p(n)}$  copies of each object  $b_j \in \mathcal{E}$  in the cell of  $\mathbf{S}(\Pi(s(u))) + cod(u)$  labelled by 0. For this purpose

- For each  $b \in \mathcal{E}$  we consider a cell in  $\mathbf{S}(\Pi(s(u))) + cod(u)$  labelled by  $l_b$  which only contains object  $\alpha_0$  initially. We also consider the following rules:
  - $[\alpha_j]_{l_b} \rightarrow [\alpha_{j+1}]_{l_b} [\alpha_{j+1}]_{l_b}$ , for  $0 \leq j \leq p(n) - 2$ .
  - $[\alpha_{p(n)-1}]_{l_b} \rightarrow [b]_{l_b} [b]_{l_b}$ .
  - $(l_b, b/\lambda, 0)$ .
- By applying the previous rules, after  $p(n)$  transition steps we get  $2^{p(n)}$  cells labelled by  $l_b$ , for each  $b \in \mathcal{E}$  in such a way that each of them contains only object  $b$ . Finally, by applying the third rule we get  $2^{p(n)}$  copies of objects  $b$  in cell 0, for each  $b \in \mathcal{E}$ .

Therefore, after the execution of  $p(n) + 1$  transition steps in each computation of  $\mathbf{S}(\Pi(s(u))) + cod(u)$  in cell 0 of the corresponding configuration, we have  $2^{p(n)}$  copies of each object  $b_1, \dots, b_m \in \mathcal{E}$ . This number of copies is enough to simulate any computation  $\mathcal{C}$  of  $\Pi(s(u)) + cod(u)$  through the system  $\mathbf{S}(\Pi(s(u)) + cod(u))$ .

From Proposition 6.6 we deduce that the family  $\{\mathbf{S}(\Pi(n)) \mid n \in N\}$  solves  $X$  in polynomial time according to Definition 5.1. Hence,  $X \in \widehat{\mathbf{PMC}}_{TDC(k+1)}$ . □

## 8 Conclusions and Further Works

The efficiency of cell-like P systems for solving **NP**-complete problems has been widely studied. The space-time tradeoff method is used to efficiently solve **NP**-complete problems in the framework of *Membrane Computing*. Membrane division, membrane creation, and membrane separation are three efficient ways to obtain exponential workspace in polynomial time. Cell division were introduced [5] into tissue-like P systems, and a linear time solution for SAT problem by tissue P systems with cell division was given [5].

In the framework of tissue P systems, there is an additional advantage when cell division is used to generate exponential workspace in polynomial time: all the

other objects in the cell are duplicated except the object that activate the cell division operation.

In this paper, the computational efficiency of tissue P systems with cell division and *without environment* has been studied. We conclude that the environment of tissue P systems can be removed without a loss of efficiency.

For future work, we plan to do further research in the study of tissue P systems with cell separation. Let us recall that, in this kind of systems, the application of separation rules only duplicates the cell while the objects are not replicated. They are simply distributed according to a prefixed criterion.

## Acknowledgements

The work was supported by Project TIN2009-13192 of the Ministerio de Ciencia e Innovación of Spain and Project of Excellence with *Investigador de Reconocida Valía*, from Junta de Andalucía, grant P08 – TIC 04200.

## References

1. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J. and Romero-Campero, F.J. A linear solution for QSAT with Membrane Creation. *Lecture Notes in Computer Science* **3850**, (2006), 241–252.
2. Pan, L. and Ishdorj, T.-O. P systems with active membranes and separation rules. *Journal of Universal Computer Science*, **10**, 5, (2004), 630–649.
3. Păun, Gh. Attacking NP-complete problems. In *Unconventional Models of Computation, UMC'2K* (I. Antoniou, C. Calude, M. J. Dinneen, eds.), Springer-Verlag, 2000, 94–115.
4. Păun, Gh. *Membrane Computing. An Introduction*. Springer-Verlag, Berlin, (2002).
5. Păun, Gh., Pérez-Jiménez, M.J. and Riscos-Núñez, A. Tissue P System with cell division. In *J. of Computers, communications & control*, **3**, 3, (2008), 295–303.
6. Pérez-Jiménez, M.J., Romero-Jiménez, A. and Sancho-Caparrini, F. Complexity classes in models of cellular computing with membranes. *Natural Computing*, **2**, 3 (2003), 265–285.
7. Pérez-Jiménez, M.J., Romero-Jiménez, A. and Sancho-Caparrini, F. A polynomial complexity class in P systems using membrane division. *Journal of Automata, Languages and Combinatorics*, **11**, 4, (2006), 423–434.