An Optimal Frontier of the Efficiency of Tissue P Systems with Cell Division

A.E. Porreca[1], Niall Murphy[2,3], Mario J. Pérez-Jiménez[4]

[1] Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano–Bicocca
Viale Sarca 336/14, 20126 Milano, Italy
[2] Departamento de Inteligencia Artificial
Universidad Politécnica de Madrid
Campus de Montegancedo s/n, Boadilla del Monte, 28660 Madrid, Spain
[3] CEI Campus Moncloa, UCM-UPM, Madrid, Spain
[4] Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
E-mail: porreca@disco.unimib.it, niall.murphy@upm.es, marper@us.es

Summary. In the framework of tissue P systems with cell division, the length of communication rules provides a frontier for the tractability of decision problems. On the one hand, the limitation on the efficiency of tissue P systems with cell division and communication rules of length 1 has been established. On the other hand, polynomial time solutions to **NP**–complete problems by using families of tissue P systems with cell division and communication rules of length at most 3 has been provided.

In this paper, we improve the previous result by showing that the HAM-CYCLE problem can be solved in polynomial time by a family of tissue P systems with cell division by using communication rules with length at most 2. Hence, a new tractability boundary is given: passing from 1 to 2 amounts to passing from non–efficiency to efficiency, assuming that $\mathbf{P} \neq \mathbf{NP}$.

1 Preliminaries

An *alphabet*, Σ, 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 Σ, 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 λ. The set of all strings over an alphabet Σ is denoted by Σ^*. In algebraic terms, Σ^* is the free monoid generated by Σ under the operation of concatenation. Subsets, finite or infinite, of Σ^* are referred to as *languages* over Σ.

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

A *multiset* m over a set A is a pair (A, f) where $f : A \to \mathbb{N}$ is a mapping. If $m = (A, f)$ is a multiset then its *support* is defined as $supp(m) = \{x \in A \mid f(x) > 0\}$. 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 $supp(m) = \{a_1, \ldots, a_k\}$, then it will be denoted as $m = \{a_1^{f(a_1)}, \ldots, a_k^{f(a_k)}\}$. 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 $m = \{a_1^{f(a_1)}, \ldots, a_k^{f(a_k)}\}$ can also be represented by the string $a_1^{f(a_1)} \ldots a_k^{f(a_k)}$ over the alphabet $\{a_1, \ldots, a_k\}$. 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:

$$A \setminus B = \{x \in A \mid \ x \notin B\}$$

In what follows, we assume the reader is already familiar with the basic notions and terminology of P systems. For details, see [9].

2 Introduction

Several different models of cell-like P systems have been successfully used to solve computationally hard problems efficiently by trading space for time. An exponential workspace is created in polynomial time by using some kind of rules, and then massive parallelism is used to simultaneously check all the candidate solutions. Inspired by living cells, several ways for obtaining exponential workspace in polynomial time were proposed: membrane division (*mitosis*) [8], membrane creation (*autopoiesis*) [4], and membrane separation (*membrane fission*) [6]. These three ways have given rise to the following models: *P systems with active membranes*, *P systems with membrane creation*, and *P systems with membrane separation*.

A new type of P systems, the so-called *tissue P systems*, was considered in [5]. Instead of considering a hierarchical arrangement, membranes/cells are placed in the nodes of a virtual graph. This variant has two biological justifications: intercellular communication and cooperation between neurons. The common mathematical model of these two mechanisms is a net of processors dealing with symbols

and communicating these symbols along channels specified in advance. Communication among cells is based on symport/antiport rules, which were introduced to P systems in [10]. One of the most interesting variants of tissue P systems was presented in [11], where the definition of tissue P systems is combined with aspects of the definition of P systems with active membranes, yielding *tissue P systems with cell division*. In these models [11], cells may replicate, that is, the two new cells generated by a division rule have exactly the same objects except for at most one differing pair of objects.

2.1 Tissue P Systems with communication rules

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

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

A *tissue P system with symport/antiport rules* $\Pi = (\Gamma, \mathcal{E}, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$, of degree $q \geq 1$ can be viewed as a set of q cells, labelled by $1, \ldots, q$, with an environment labelled by 0 such that: (a) $\mathcal{M}_1, \ldots, \mathcal{M}_q$ are strings over Γ representing the finite multisets of objects (elements in Γ) initially placed in the q cells of the system; (b) $\mathcal{E}$ is the set of objects located initially in the environment of the system, all of them appearing in an *arbitrary number of copies*; and (c) $i_{out} \in \{0, 1, 2, \ldots, q\}$ represents a distinguished cell or the environment which will encode the output of the system.

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 $|u| + |v|$, that is, the total number of objects which appear in the rule.

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 systems has an underlying directed graph whose nodes are the cells of the system and the arcs are obtained from communication rules. In this context, the environment can be considered as a virtual node of the graph such that their connections are defined by 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 is described by all multisets of objects over Γ associated with all the cells present in the system, and the multiset of objects over $\Gamma - \mathcal{E}$ 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 $(\mathcal{M}_1, \cdots, \mathcal{M}_q; \emptyset)$. A configuration is a *halting configuration* if no rule of the system is applicable to it.

Let us fix a tissue P system with symport/antiport rules Π. 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 Π is a (finite or infinite) sequence of configurations such that:

1. the first term of the sequence is an 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 (a cell or the environment) i_{out} in the halting configuration.
Notation: If $\mathcal{C} = \{\mathcal{C}_i\}_{i<r+1}$ $(r \in \mathbf{N})$ is a halting computation of Π, 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 $\mathcal{C}_i$.

2.2 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 active membranes [8]. 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 2.2 *A tissue P system with cell division of degree $q \geq 1$ is a tuple $\Pi = (\Gamma, \mathcal{E}, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$, where:*

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

A *tissue P system with cell division* is a *tissue P system with symport/antiport rules* where division rules of cells are allowed.

When applying a division rule $[a]_i \to [b]_i[c]_i$, 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 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 systems 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 rule can be added), with the following important remark: if a cell divides, only the division rule is applied to that cell at that step; the objects inside that cell do not evolve by means of communication rules. In other words, we can think that before division a cell interrupts all its communication channels with the other cells and with the environment. The new cells resulting from division will only interact with other cells or with the environment at the next step – providing they do not divide once again. The label of a cell identifies the rules which can be applied to it precisely.

2.3 Recognizer Tissue P Systems with Cell Division

Let us recall that a *decision problem* is a pair (I_X, θ_X) where I_X is a language over a finite alphabet (whose elements are called *instances*) and θ_X is a total boolean function over I_X. There are many different ways to describe instances of a decision problem, but we assume that each problem has associated with it a fixed *reasonable encoding scheme* (in the sense of [2], page 10) which provides a string associated

with each problem instance. The *size* of an instance $u \in I_X$ is the length of the string associated with it by means of a reasonable encoding scheme.

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 over a finite alphabet, 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 Σ, its associated decision problem is $X_L = (I_{X_L}, \theta_{X_L})$, where $I_{X_L} = \Sigma^*$, 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 by means of languages recognizer devices.

In order to study the computational efficiency of membrane systems, the notions from classical *computational complexity theory* are adapted for Membrane Computing, and a special class of cell-like P systems is introduced in [13]: *recognizer P systems* (called *accepting P systems* in a previous paper [12]). Similarly, *recognizer tissue P systems* are introduced in [11].

Definition 2.3 *A* recognizer tissue P system with cell division *of degree $q \geq 1$ is a tuple $\Pi = (\Gamma, \Sigma, \mathcal{E}, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{in}, i_{out})$, where:*

1. *$(\Gamma, \mathcal{E}, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$ is a tissue P system with cell division of degree $q \geq 1$ (as defined in the previous section).*
2. *The working alphabet Γ has two distinguished objects* `yes` *and* `no` *being, at least, one copy of them present in some initial multisets $\mathcal{M}_1$, …, $\mathcal{M}_q$, but none of them are present in $\mathcal{E}$.*
3. *Σ is an (input) alphabet strictly contained in Γ such that $\mathcal{E} \cap \Sigma = \emptyset$.*
4. *$\mathcal{M}_1, \ldots, \mathcal{M}_q$ are strings over $\Gamma \setminus \Sigma$.*
5. *$i_{in} \in \{1, \ldots, q\}$ is the input cell.*
6. *$i_{out} = 0$, that is, the output region is the environment.*
7. *All computations halt.*
8. *If $\mathcal{C}$ is a computation of Π, then either object* `yes` *or object* `no` *(but not both) must have been released into the environment, and only at the last step of the computation.*

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

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

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

where Ψ is the Parikh function, and $M_{i,0}$ is the multiset over $\Gamma \setminus \mathcal{E}$ associated with the environment at configuration $\mathcal{C}_i$. In particular, $M_{r,0}$ is the multiset over $\Gamma \setminus \mathcal{E}$ associated with the environment at the halting configuration $\mathcal{C}_r$.

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

For each natural number $k \geq 1$, we denote by $\mathbf{TDC}(k)$ the class of recognizer tissue P systems with cell division and communication rules with length at most k.

2.4 Polynomial Complexity Classes of Tissue P systems with Cell Division

Now, we define what it means to solve a decision problem in the framework of tissue P systems efficiently and in a uniform way. Since we define each tissue P system to work on a finite number of inputs, to solve a decision problem we define a numerable family of tissue P systems.

Definition 2.4 *We say that a decision problem $X = (I_X, \theta_X)$ is solvable in a uniform way and polynomial time by a family $\mathbf{\Pi} = \{\Pi(n) \mid n \in \mathbb{N}\}$ of recognizer tissue P systems with cell division if the following holds:*

1. *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 $n \in \mathbb{N}$.*
2. *There exists a pair (cod, s) of polynomial-time computable functions over I_X such that:*
   (a) *for each instance $u \in I_X$, $s(u)$ is a natural number[5] and $cod(u)$ is an input multiset of the system $\Pi(s(u))$;*
   (b) *for each $n \in \mathbb{N}$, $s^{-1}(n)$ is a finite set;*
   (c) *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;*
   (d) *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 $\theta_X(u) = 1$;*

[5] Note, for this definition to be compatible with the notion of uniformity in Boolean circuit complexity [15] we restrict $s(u)$ to be some function on $|u|$, the length of u.

(e) 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.*

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 $\mathbf{PMC_R}$ 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 $\mathbf{PMC_R}$ is closed under complement and polynomial–time reductions [12].

3 Computational Efficiency of Tissue P Systems with Cell Division

It is well known that tissue P systems with cell division are able to solve computationally hard problems efficiently. Specifically, **NP**–complete problems have been solved in linear time [1] by using families of tissue P systems with cell division and communication rules of length at most 3. Thus, $\mathbf{NP} \cup \mathbf{co-NP} \subseteq \mathbf{PMC}_{TDC(3)}$. In [3] has been proved $\mathbf{P} = \mathbf{PMC}_{TDC(1)}$, that is, only tractable problems can be efficiently solved by using families of tissue P systems with cell division and communication rules of length 1. Therefore, in the framework of tissue P systems with cell division, passing the maximum length of communication rules of the systems from 1 to 3 amounts to passing from non–efficiency to efficiency, assuming that $\mathbf{P} \neq \mathbf{NP}$. An interesting challenge is to provide new efficient solutions to computationally hard problems by means of tissue P systems with cell division by using communication rules of length at most 2.

In the next Section, we give a family of tissue P systems with cell division and communication rules of length at most 2 which solves the `HAM-CYCLE` problem, a well known **NP**–complete problem, in polynomial time.

4 On efficiency of TDC(2)

We start by giving some concepts and notations related to graph theory that we will use throughout this paper.

4.1 Hamiltonian cycles in directed graphs

First of all, let us recall some concepts related to graph theory which are relevant in this paper.

Definition 4.1 *Let* $G = (V, E)$ *be a directed graph. Let* $V = \{1, \ldots, n\}$*,* $E = \{(u_1, v_1), \ldots, (u_p, v_p)\} \subset V \times V$*. A finite sequence* $\gamma = (u_{\alpha_1}, u_{\alpha_2}, \ldots, u_{\alpha_r}, u_{\alpha_{r+1}})$ *of nodes of* G *is a simple path of* G *of length* $r \geq 1$ *if the following holds:*

- $\forall i\ (1 \leq i \leq r \rightarrow (u_{\alpha_i} u_{\alpha_{i+1}}) \in E)$.
- $|\{u_{\alpha_1}, u_{\alpha_2}, \ldots, u_{\alpha_r}\}| = r$.

If $u_{\alpha_{r+1}} \notin \{u_{\alpha_1}, u_{\alpha_2}, \ldots, u_{\alpha_r}\}$, then we say that γ is a simple path of length r from u_{α_1} to $u_{\alpha_{r+1}}$. If $u_{\alpha_{r+1}} = u_{\alpha_1}$, then we say that γ is a simple cycle of length r (in this case, we assume $r \geq 2$). A Hamiltonian path of G from $a \in V$ to $b \in V$ $(a \neq b)$ is a simple path $\gamma = (u_{\alpha_1}, u_{\alpha_2}, \ldots, u_{\alpha_r}, u_{\alpha_{r+1}})$ from a to b such that $a = u_{\alpha_1}$, $b = u_{\alpha_{r+1}}$, and $V = \{u_{\alpha_1}, u_{\alpha_2}, \ldots, u_{\alpha_r}, u_{\alpha_{r+1}}\}$. A Hamiltonian cycle of G is a simple cycle $\gamma = (u_{\alpha_1}, u_{\alpha_2}, \ldots, u_{\alpha_r}, u_{\alpha_{r+1}})$ of G such that $V = \{u_{\alpha_1}, u_{\alpha_2}, \ldots, u_{\alpha_r}\}$.

If $\gamma = (u_{\alpha_1}, u_{\alpha_2}, \ldots, u_{\alpha_r}, u_{\alpha_{r+1}})$ is a simple path of G then we also denote it by the set $\{(u_{\alpha_1}, u_{\alpha_2})_1, (u_{\alpha_2}, u_{\alpha_3})_2, \ldots, (u_{\alpha_r}, u_{\alpha_{r+1}})_r\}$. That is, $(u_{\alpha_k}, u_{\alpha_{k+1}})_k$ can be interpreted as the k-th arc of the path γ, for each k $(1 \leq k \leq r)$.

Given a directed graph $G = (V, E)$, throughout this paper we denote

$$A_G = \{(u,v)_k \mid u, v, k \in \{1, \ldots, n\} \ \wedge \ (u,v) \in E\}$$
$$A'_G = \{(u,v)'_k \mid u, v, k \in \{1, \ldots, n\} \ \wedge \ (u,v) \in E\}$$
$$A''_G = \{(u,v)''_k \mid u, v, k \in \{1, \ldots, n\} \ \wedge \ (u,v) \in E\}$$

Proposition 4.2 *Let $G = (V, E)$ be a directed graph. Let $V = \{1, \ldots, n\}$ and $A_G = \{(u,v)_k |\ u, v, k \in \{1, \ldots, n\} \ \wedge \ (u,v) \in E\}$. If $B \subseteq A_G$ then the following assertions are equivalent:*

1. *B is a Hamiltonian cycle.*
2. *$|B| = n$ and the following holds: for each $\forall u, u', v, v', k, k' \in \{1, \ldots, n\}$,*
   (a) $[(u,v)_k \in B \wedge (u',v')_{k'} \in B \wedge (u,v)_k \neq (u',v')_{k'} \rightarrow k \neq k']$
   (b) $[(u,v)_k \in B \wedge (u',v')_{k'} \in B \wedge (u,v)_k \neq (u',v')_{k'} \rightarrow u \neq u']$
   (c) $[(u,v)_k \in B \wedge (u',v')_{k'} \in B \wedge (u,v)_k \neq (u',v')_{k'} \rightarrow v \neq v']$
   (d) $[(u,v)_k \in B \wedge (u',v')_{k+1} \in B \rightarrow v = u']$

Proof: Let $B = \{(u_{\alpha_1}, u_{\alpha_2})_1, (u_{\alpha_2}, u_{\alpha_3})_2 \ldots, (u_{\alpha_m}, u_{\alpha_{r+1}})_n\}$ be a Hamiltonian cycle of G. Then, $|B| = n$ and the conditions (a), (b), (c) and (d) from (2) hold.

Let $B \subseteq A_G$ such that $|B| = n$ and the conditions (a), (b), (c) and (d) from (2) hold. Then, from (a) the set B must to be of the form

$$B = \{(u_{\alpha_1}, v_{\alpha_1})_1, (u_{\alpha_2}, v_{\alpha_2})_2 \ldots, (u_{\alpha_n}, v_{\alpha_n})_n\}$$

where:

- From (d) we deduce that $\forall i\ (1 \leq i \leq n-1 \rightarrow v_{\alpha_i} = u_{\alpha_{i+1}})$.
- From (b) we have $V = \{u_{\alpha_1}, u_{\alpha_2}, \ldots, u_{\alpha_n}\}$.

Finally, on the one hand we have $v_{\alpha_n} \in \{u_{\alpha_1}, u_{\alpha_2} \ldots, u_{\alpha_n}\}$. On the other hand, by condition (c) we deduce that $v_{\alpha_n} \notin \{v_{\alpha_1}, \ldots, v_{\alpha_{n-1}}\} = \{u_{\alpha_2}, \ldots, u_{\alpha_n}\}$. Thus $v_{\alpha_n} = u_{\alpha_1}$.

□

Remark 1: If $B \subseteq A_G$ is a Hamiltonian cycle of G, then it cannot have different pairs of elements of the types $(i,j)_k$ and $(i,j')_{k'}$, or of the types $(i,j)_k$ and $(i',j)_{k'}$, or $(i,j)_k$ and $(i',j')_k$, or $(i,j)_k$ and $(i',j')_{k+1}$ with $j \neq i'$.

Remark 2: Let us notice that if $(u_{\alpha_1}, u_{\alpha_2}, \ldots, u_{\alpha_n}, u_{\alpha_1})$ is a Hamiltonian cycle of G of length n, then we can describe it by the following subset of A_G:

$$B_1 = \{(u_{\alpha_1}, u_{\alpha_2})_1, (u_{\alpha_2}, u_{\alpha_3})_2, \ldots, (u_{\alpha_n}, u_{\alpha_1})_n\}$$

But $(u_{\alpha_2}, u_{\alpha_3}, \ldots, u_{\alpha_m}, u_{\alpha_1}, u_{\alpha_2})$ is also a Hamiltonian cycle of G of length m. It can be described as follows:

$$B_2 = \{(u_{\alpha_2}, u_{\alpha_3})_1, (u_{\alpha_3}, u_{\alpha_4})_2, \ldots, (u_{\alpha_1}, u_{\alpha_2})_n\}$$

Thus, given a Hamiltonian cycle γ of G, there are exactly n different subsets of A_G codifying exactly the cycle γ.
Remark 3: Let us supose that the total number of Hamiltonian cycles of G is q. Then, the number of different subsets B of A_G verifying conditions (a), (b), (c), and (d) of the previous Proposition is exactly $n \cdot q$.

4.2 An efficient, uniform solution of HAM-CYCLE in TDC(2)

In this Section we provide a uniform and polynomial time solution for the `HAM-CYCLE` problem by using a family of tissue P systems with cell division and communication rules of length at most 2.

Let us recall that the `HAM-CYCLE` problem is the following: *given a directed graph, to determine whether or not there exists a Hamiltonian cycle in the graph.* This is a well known **NP**-complete problem [2].

The proposed solution follows a brute force algorithm implemented in the framework of recognizer tissue P systems with cell division. The solution consists of the following stages:

- *Generation Stage*: From the input cell labelled by *in*, all possible combinations of arcs including a code of their position in potential paths, are generated in those cells and by using cell division in an adequate way.
- *Checking Stage*: In each cell labelled by *in*, it is checked whether or not the different combinations of arcs encode Hamiltonian cycles of the graph.
- *Output Stage*: The system sends the right answer to the environment according to the results of the previous stage.

Then, we define a family $\mathbf{\Pi} = \{\Pi(n) : n \in \mathbb{N}\}$ of recognizer tissue P system with cell division from $\mathbf{TDC}(2)$, such that each system $\Pi(n)$ will process all instances G of `HAM-CYCLE` with n nodes.

For each $n \in \mathbb{N}$, we consider the recognizer tissue P system with cell division from $\mathbf{TDC}(2)$,

$$\begin{aligned}\Pi(n) = (\Gamma, \Sigma, \mathcal{E}, \mathcal{M}_{in}, \mathcal{M}_h, \mathcal{M}_y, \mathcal{M}_{yes}, \mathcal{M}_{no}, \mathcal{M}_{out},\\ \mathcal{M}_{e_{i,j,k}}(1 \leq i, j, k \leq n), \mathcal{M}_{c_i}(1 \leq i, \leq n), \mathcal{R}, i_{in}, i_{out})\end{aligned}$$

defined as follows:

- The input alphabet is $\Sigma = \{(i,j)_k \mid 1 \leq i,j,k \leq n\}$.
- The working alphabet is

$$\begin{aligned}\Gamma = &\{(i,j)_k, (i,j)'_k, (i,j)''_k, (i,j)_{k,r}, (i,j)'_{k,r}, (i,j)''_{k,r} \mid 1 \leq i,j,k \leq n \ \wedge \ 1 \leq r \leq n^3\} \cup \\ &\{w_i \mid 1 \leq i \leq n^3+6\} \cup \{c_r, h_r, y_r \mid 1 \leq r \leq n^3\} \cup \\ &\{w, c, c', c'', h, h', h'', h''', y, y'y'', y''', y'''', x, yes, no, \#\}\end{aligned}$$

- The alphabet of the environment is:

$$\mathcal{E} = \{w_i \mid 1 \leq i \leq n^3+5\} \ \cup \ \{w, c'', y'', h'', y''', h''', y''''\}$$

- Initial multisets:

$$\begin{cases} \mathcal{M}_{in} = c^n \ y \ h \\ \mathcal{M}_{e_{i,j,k}} = (i,j)''_{k,n^3}, \ 1 \leq i,j,k \leq n \\ \mathcal{M}_{c_i} = c_{n^3}, \ 1 \leq i \leq n \\ \mathcal{M}_h = h_{n^3} \\ \mathcal{M}_y = y_{n^3} \\ \mathcal{M}_{yes} = yes \\ \mathcal{M}_{no} = w_{n^3+6} \ no \\ \mathcal{M}_{out} = x \end{cases}$$

- The set R of rules consists of the following rules:

(1) $(no \, , \, w_r \, / \, w_{r-1} \, , \, 0)$, for $2 \leq r \leq n^3+6$.
(2) $(no \, , \, w_1 \, / \, w \, , \, 0)$.
(3) $[\,(i,j)_k\,]_{in} \rightarrow [\,(i,j)'_k\,]_{in} \ [\,\#\,]_{in}$, for $1 \leq i,j,k \leq n$.
(4) $[\,(i,j)''_{k,r}\,]_{e_{i,j,k}} \rightarrow [\,(i,j)''_{k,r-1}\,]_{e_{i,j,k}} \ [\,(i,j)''_{k,r-1}\,]_{e_{i,j,k}}$, for $1 \leq i,j,k \leq n$ and $2 \leq r \leq n^3$.
(5) $[\,(i,j)''_{k,1}\,]_{e_{i,j,k}} \rightarrow [\,(i,j)''_k\,]_{e_{i,j,k}} \ [\,(i,j)''_k\,]_{e_{i,j,k}}$, for $1 \leq i,j,k \leq n$.
(6) $[\,c_r\,]_{c_i} \rightarrow [\,c_{r-1}\,]_{c_i} \ [\,c_{r-1}\,]_{c_i}$, for $1 \leq i \leq n \ \wedge \ 1 \leq r \leq n^3$.
(7) $[\,y_r\,]_y \rightarrow [\,y_{r-1}\,]_y \ [\,y_{r-1}\,]_y$, for $1 \leq r \leq n^3$.
(8) $[\,h_r\,]_h \rightarrow [\,h_{r-1}\,]_h \ [\,a_{r-1}\,]_h$, for $1 \leq r \leq n^3$.
(9) $(in \, , \, (i,j)'_k \, / \, (i,j)''_k \, , \, e_{i,j,k})$, for $1 \leq i,j,k \leq n$.
(10) $(in \, , \, c \, / \, c' \, , \, c_i)$, for $1 \leq i \leq n$.
(11) $(in \, , \, y \, / \, y' , \, y)$.
(12) $(in \, , \, h \, / \, h' , \, h)$.
(13) $(in, \, (i,j)''_k \, (i,j')''_{k'} \, / \, \lambda, \, 0)$, for $1 \leq i,j,j',k,k' \leq n$.
(14) $(in, \, (i,j)''_k \, (i',j)''_{k'} \, / \, \lambda, \, 0)$, for $1 \leq i,i',j,k,k' \leq n$.
(15) $(in, \, (i,j)''_k \, (i',j')''_{k+1} \, / \, \lambda, \, 0)$, for $1 \leq i,i',j,j',k \leq n$, and $j \neq i'$.
(16) $(in, \, (i,j)''_k \, (i',j')''_k \, / \, \lambda, \, 0)$, for $1 \leq i,i',j,j',k \leq n$.
(17) $(in \, , \, c' \, / \, c'' \, , \, 0)$.
(18) $(in \, , \, y' \, / \, y'' \, , \, 0)$.
(19) $(in \, , \, h' \, / \, h'' \, , \, 0)$.
(20) $(in \, , \, (i,j)''_k \, c'' \, / \, \lambda \, , \, 0)$ for $1 \leq i,j,k \leq n$.
(21) $(in \, , \, y'' \, / \, y''' \, , \, 0)$.
(22) $(in \, , \, h'' \, / \, h''' \, , \, 0)$.

(23) $(in\,,\, c''\,h'''\,/\,\lambda\,,\, 0)$.
(24) $(in\,,\, y'''\,/\,y''''\,,\, 0)$.
(25) $(in\,,\, h'''\,y''''\,/\,\lambda\,,yes)$.
(26) $(yes\,,\, y''''\,yes\,/\,\lambda\,,\, out)$.
(27) $(out\,,\, x\,yes\,/\,\lambda\,,\, 0)$.
(28) $(no\,,\, w\,no\,/\,\lambda\,,\, out)$.
(29) $(out\,,\, x\,no\,/\,\lambda\,,\, 0)$.

- The input cell is $i_{in} = in$.
- The output region is the environment, $i_{out} = 0$.

4.3 An Overview of the Computations

A family of recognizer tissue P systems with cell division is constructed above. Let $G = (V, E)$, with $V = \{1, \ldots, n\}$ and $E = \{(u_1, v_1), \ldots, (u_p, v_p)\}$, be an arbitrary instance of the HAM-CYCLE problem.

The *size* mapping[6]on the set of instances is defined as $s(G) = n$, and the encoding of the instance is the multiset

$$cod(G) = \{(u_i, v_i)_k \mid 1 \leq i \leq p \ \wedge \ 1 \leq k \leq n \ \wedge \ (u_i, v_i) \in E\}$$

That is, $(u_i, v_i)_k$ denotes arc (u_i, v_i) "placed" in "position k". Then the graph G will be processed by system $\Pi(s(G))$ with input multiset $cod(G)$.

Then, we informally describe how system $\Pi(s(G))$ with input multiset $cod(G)$ works, in order to process the instance G of the HAM-CYCLE problem.

At the initial configuration of $\Pi(s(G)) + cod(G)$ we have the following:

- n copies of object c, objects y, h, and $(u_i, v_i)_j$, for $(u_i, v_i) \in E$, $1 \leq k \leq n$, in cell labelled by in,
- Objects $(i, j)''_{k,n^3}$ in cell labelled by $e_{i,j,k}$.
- Objects c_{n^3} in each cell labelled by c_i, for $1 \leq i \leq n$.
- Object h_{n^3} in cell labelled by h, object y_{n^3} in cell labelled by y, object yes in cell labelled by h, objects no and w_{n^3+6} in cell labelled by no, and object x in cell labelled by out.

Let us start with the **generation stage**. This stage spends n^3 steps. At this stage, we try to generate all the possible subsets of arcs of the graph which contain their potential positions in a path according the notations introduced in Section 4.1 (in fact, subsets of A'_G).

If $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \ldots)$ be a computation of the tissue P system $\Pi(n)$, then at configuration $\mathcal{C}_{n^3}$:

[6] Note, for this family to be considered uniform in the sense of Boolean circuit families [15] we may modify s so that its mapping to the number n depends only on the length of of G. For example, if the graph G is encoded as a binary adjacency matrix, then $s(u) = \sqrt{|u|} = n$.

1. There are $2^{n \cdot p}$ cells labelled by in such that each of them contains a different subset of $A'_G = \{(u_i, v_i)'_k \mid 1 \leq i \leq p \wedge 1 \leq k \leq n \ \wedge \ (u_i, v_i) \in E\}$ as well as object y, object h and n copies of object c.
2. For each i, j, k $(1 \leq i, j, k \leq n)$ there are 2^{n^3} cells labelled by $e_{i,j,k}$, each of them only containing object $(i, j)''_k$.
3. For each i $(1 \leq i \leq n)$ there are 2^{n^3} cells labelled by c_i, 2^{n^3} cells labelled by h, and 2^{n^3} cells labelled by y, only containing object c', object h', object y' respectively.
4. There is a cell labelled by no, a cell labelled by yes and a cell labelled by out such that $\mathcal{C}_{n^3}(no) = \{w_6, no\}$, $\mathcal{C}_{n^3}(yes) = \{yes\}$, $\mathcal{C}_{n^3}(out) = \{x\}$.

Now, the **checking stage** starts. This stage spends 3 steps. At this stage, we try to determine whether or not there exists a cell labelled by in which contains a subset of A''_G that encodes a Hamiltonian cycle of G. For that purpose, we will use rules of types (13), (14), (15), and (16) in order to select possible paths of the graph. After that, rules of type (20), (21), and (22) allow us to determine cells labelled by in at configuration $\mathcal{C}_{n^3+3}$ which encode Hamiltonian cycles.

Finally, the **output stage** spends 3 steps if the answer is affirmative and 4 if it is negative. At configuration $\mathcal{C}_{n^3+3}$, the existence of Hamiltonian cycles in the graph is characterized by the absence of objects c'' in some cell labelled by in. At configuration $\mathcal{C}_{n^3+4}$, the previous condition is expressed by the existence of some cell labelled by in which contains object h'''. At configuration $\mathcal{C}_{n^3+5}$, the existence of Hamiltonian cycles in the graph is characterized by the presence of some object y'''' in cell labelled by yes. Rules of type (26), (27), (28), and (29) produce the right answer.

5 A Formal Verification

The aim of this section is to present a formal proof on the fact that the family of recognizer tissue P systems with cell division constructed in the previous section solves the `HAM-CYCLE` problem in a uniform way and polynomial time, according to Definition 2.4.

5.1 Polynomial Uniformity of the Family

Then, we will show that the family $\mathbf{\Pi} = \{\Pi(n) \mid n \in \mathbb{N}\}$ defined above is polynomially uniform by Turing machines. To this aim we prove that $\Pi(n)$ is built in polynomial time with respect to the number of nodes of the instance G of the `HAM-CYCLE` problem.

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

1. Size of the alphabet: $3n^4 + 7n^3 + 23 \in \Theta(n^4)$.

2. Initial number of cells: $n^3 + n + 6 \in \Theta(n^3)$.
3. Initial number of objects: $|E| + n^3 + 2n + 8 \in \Theta(n^3)$.
4. Number of rules: $n^6 + 4n^5 - n^4 + 7n^3 + n + 20n \in \Theta(n^6)$.
5. Maximal length of a rule: $2 \in \Theta(1)$.

Therefore, there exists a deterministic Turing machine that builds the system $\Pi(n)$ in time polynomial with respect to n.

5.2 Soundness and Completeness of the Family

In order to show the soundness and completeness of the family $\mathbf{\Pi}$ with respect to $(\texttt{HAM-CYCLE}, cod, s)$, we describe the full contents of any cells in any instant of each computation of the tissue $\Pi(s(G)) + cod(G)$ that processes instance G.

Theorem 5.1 *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ be a computation of the tissue P system $\Pi(n)$. For every t $(1 \leq t \leq n \cdot p)$, the configuration $\mathcal{C}_t$ verifies the following properties:*

(1) *There are 2^t cells labelled by in such that each of them contains a different subset of $A'_G = \{(u_i, v_i)'_k \mid 1 \leq i \leq p \wedge 1 \leq k \leq n \ \wedge \ (u_i, v_i) \in E\}$ of size lower than or equal than t, as well as object y, object h and n copies of object c.*

(2) *For each i, j, k $(1 \leq i, j, k \leq n)$ there are 2^t cells labelled by $e_{i,j,k}$ each of them only containing object $(i, j)''_{k,n^3-t}$.*

(3) *For each i $(1 \leq i \leq n)$ there are 2^t cells labelled by c_i, 2^t cells labelled by h, and 2^t cells labelled by y, only containing object c_{n^3-t}, object h_{n^3-t}, object y_{n^3-t} respectively.*

(4) *There is a cell labelled by no, a cell labelled by yes and a cell labelled by out such that $\mathcal{C}_t(no) = \{w_{n^3-t+6}, no\}$, $\mathcal{C}_t(yes) = \{yes\}$, $\mathcal{C}_t(out) = \{x\}$.*

Proof: By induction on t. Let us start analyzing the basic case $t = 1$.

(1) At the first step of computation $\mathcal{C}$, a rule of the form $[\,(u_0, v_0)_{k_0}\,]_{in} \to [\,(u_0, v_0)'_{k_0}\,]_{in}\ [\,\#\,]_{in}$, with $(u_0, v_0) \in E$, will be applied to cell labelled by in. Then, two new cells labelled by in will be created, each of them containing a subset of A'_G of size lower than or equal to 1: one is $(u_0, v_0)'_{k_0}$ and the another is $\emptyset$. The initial objects y, h and n copies of object c remain unchanged.

(2) For each i, j, k $(1 \leq i, j, k \leq n)$, at the first step of computation $\mathcal{C}$, the rule $[\,(i, j)''_{k,n^3}\,]_e \to [\,(i, j)''_{k,n^3-1}\,]_e\ [\,(i, j)''_{k,n^3-1}\,]_e$ will be applied to cell labelled by $e_{i,j,k}$. Then, two new cells labelled by $e_{i,j,k}$ will be created, each of them only containing object $(i, j)''_{k,n^3-1}$.

(3) For each i $(1 \leq i \leq n)$, at the first step of computation $\mathcal{C}$, a rule of the type (6) is applied to cell labelled by c_i. It produces two new cells labelled by c_i, each of them only containing object c_{n^3-1}.
At the first step of computation $\mathcal{C}$, a rule of the type (7) is applied to the cell labelled by y, and a rule of the type (8) is applied to cell labelled by h. They produce two new cells labelled by y, each of them only containing object y_{n^3-1}, and two new cells labelled by h each of them only containing the object h_{n^3-1}.

(4) At the first step of computation $\mathcal{C}$, a rule of the type (1) is applied to the cell labelled by *no*. The new content of that cell labelled by *no* is $\{w_{n^3-1}\, no\}$. No rule is applied neither to cell *yes* nor cell *out*.

By induction hypothesis, let t be such that $1 \leq t < n \cdot p$ and let us suppose the result holds for t. Let us see that the result also holds for $t+1$. For this purpose, let us notice that configuration $\mathcal{C}_{t+1}$ is obtained from configuration $\mathcal{C}_t$ by applying:

- A rule of the type (3) $[\,(u_0, v_0)_k\,]_{in} \rightarrow [\,(u_0, v_0)'_k\,]_{in}\ [\,\#\,]_{in}$ which is selected in a nondeterministic manner among all possible applicable rules to each cell labelled by *in* (there exist such rules because of $t < n \cdot p$).
- For each i, j, k $(1 \leq i, j, k \leq n)$, rules of the type (4) corresponding to $r = n^3 - t$ in each cell labelled by $e_{i,j,k}$:

$$[\,(i,j)''_{k,n^3-t}\,]_{e_{i,j,k}} \rightarrow [\,(i,j)''_{k,n^3-t-1}\,]_{e_{i,j,k}}\ [\,(i,j)''_{k,n^3-t-1}\,]_{e_{i,j,k}}$$

- For each i $(1 \leq i \leq n)$, rules of the type (6) corresponding to $r = n^3 - t$ in each cell labelled by c_i: $[\,c_{n^3-t}\,]_{c_i} \rightarrow [\,c_{n^3-t-1}\,]_{c_i}\ [\,c_{n^3-t-1}\,]_{c_i}$.
- Rules of the type (7) corresponding to $r = n^3 - t$ in each cell labelled by y:

$$[\,y_{n^3-t}\,]_y \rightarrow [\,y_{n^3-t-1}\,]_y\ [\,y_{n^3-t-1}\,]_y$$

- Rules of the type (8) corresponding to $r = n^3 - t$ in each cell labelled by h:

$$[\,h_{n^3-t}\,]_h \rightarrow [\,h_{n^3-t-1}\,]_h\ [\,h_{n^3-t-1}\,]_h$$

- A rule of the type (1) corresponding to $r = n^3 - t$ in cell labelled by *no*:

$$(no\,,\, w_{n^3-t} \,/\, w_{n^3-t-1}\,,\, 0)$$

Therefore, the following conclusions are reached at:

(1) By induction hypothesis, in $\mathcal{C}_t$ there are 2^t cells labelled by *in*, each of them containing object y, object h, object n copies of object c, and a different subset of A'_G with size $\leq t$. Thus, when applying a rule of the type $[\,(u_0, v_0)_k\,]_{in} \rightarrow [\,(u_0, v_0)'_k\,]_{in}\ [\,\#\,]_{in}$, with $(u_0, v_0) \in E$, we will have 2^{t+1} cells labelled by *in* such that 2^t of that cells have the same content that they had at configuration $\mathcal{C}_t$, and the rest of 2^t objects $(u_0, v_0)'_k$ are added. That is, at configuration $\mathcal{C}_{t+1}$, we will have 2^{t+1} cells labelled by *in*, each of them containing object y, object h, n copies of object c, and a different subset of A'_G with size $\leq t+1$.

(2) By induction hypothesis, for each i, j, k $(1 \leq i, j, k \leq n)$ in $\mathcal{C}_t$ there are 2^t cells labelled by $e_{i,j,k}$, each of them only containing object $(i,j)''_{k,n^3-t}$. By applying rule $[\,(i,j)''_{k,n^3-t}\,]_{e_{i,j,k}} \rightarrow [\,(i,j)''_{k,n^3-t-1}\,]_{e_{i,j,k}}\ [\,(i,j)''_{k,n^3-t-1}\,]_{e_{i,j,k}}$, we will have 2^{t+1} cells labelled by $e_{i,j,k}$, each of them only containing object $(i,j)''_{k,n^3-t-1}$.

(3) By induction hypothesis, for each i $(1 \leq i \leq n)$ in $\mathcal{C}_t$ there are 2^t cells labelled by c_i, each of them only containing object c_{n^3-t}. By applying rule $[\,c_{n^3-t}\,]_{c_i} \rightarrow [\,c_{n^3-t-1}\,]_{c_i}\ [\,c_{n^3-t-1}\,]_{c_i}$ we will have 2^{t+1} cells labelled by c_i, each of them only containing object c_{n^3-t-1}.

By induction hypothesis, in $\mathcal{C}_t$ there are 2^t cells labelled by h, each of them only containing object h_{n^3-t}, and 2^t cells labelled by y, each of them only containing object y_{n^3-t}. By applying the rules $[\,h_{n^3-t}\,]_h \to [\,h_{n^3-t-1}\,]_h\ [\,h_{n^3-t-1}\,]_h$ and $[\,y_{n^3-t}\,]_y \to [\,y_{n^3-t-1}\,]_y\ [\,y_{n^3-t-1}\,]_y$ we will have 2^{t+1} cells labelled by h, each of them only containing object h_{n^3-t-1}, and 2^{t+1} cells labelled by y, each of them only containing object y_{n^3-t-1}.

(4) By induction hypothesis, in $\mathcal{C}_t$ there is a cell labelled by *no* which contains object w_{n^3-t+6} and object *no*. By applying rule $(no\,,\, w_{n^3-t}\,/\,w_{n^3-t-1}\,,\, 0)$, it will contain object $w_{n^3-(t+1)+6}$ and object *no*. By the way, no rules are applicable to cells *yes* or *out* at configuration $\mathcal{C}_t$.

□

From the previous proposition, we can describe the configuration $\mathcal{C}_{n\cdot p}$ of each computation $\mathcal{C}$ of $\Pi(n)$.

Corollary 5.2 *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ be a computation of the tissue P system $\Pi(n)$. Configuration $\mathcal{C}_{n\cdot p}$ verifies the following:*

(1) *There are $2^{n\cdot p}$ cells labelled by in such that each of them contains a different subset of $A'_G = \{(u_i, v_i)'_k \mid 1 \le i \le p \wedge 1 \le k \le n\ \wedge\ (u_i, v_i) \in E\}$, as well as object y, object h and n copies of object c.*

(2) *For each i, j, k $(1 \le i, j, k \le n)$ there are $2^{n\cdot p}$ cells labelled by $e_{i,j,k}$ each of them only containing object $(i,j)''_{k,n^3-n\cdot p}$.*

(3) *For each i $(1 \le i \le n)$ there are $2^{n\cdot p}$ cells labelled by c_i, $2^{n\cdot p}$ cells labelled by h, and $2^{n\cdot p}$ cells labelled by y, only containing object $c_{n^3-n\cdot p}$, object $h_{n^3-n\cdot p}$, object $y_{n^3-n\cdot p}$ respectively.*

(4) *There is a cell labelled by no, a cell labelled by yes and a cell labelled by out such that $\mathcal{C}_{n\cdot p}(no) = \{w_{n^3-n\cdot p+6}, no\}$, $\mathcal{C}_{n\cdot p}(yes) = \{yes\}$, $\mathcal{C}_{n\cdot p}(out) = \{x\}$.*

Theorem 5.3 *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ be a computation of the tissue P system $\Pi(n)$. For every t $(n \cdot p + 1 \le t \le n^3)$, configuration $\mathcal{C}_t$ verifies the following:*

(1) *There are $2^{n\cdot p}$ cells labelled by in whose content is equal to the content of those cells in configuration $\mathcal{C}_{n\cdot p}$.*

(2) *For each i, j, k $(1 \le i, j, k \le n)$ there are 2^t cells labelled by $e_{i,j,k}$, each of them only containing object $(i,j)''_{k,n^3-t}$ (by considering $(i,j)''_{k,0} = (i,j)''_k$).*

(3) *For each i $(1 \le i \le n)$ there are 2^t cells labelled by c_i, 2^t cells labelled by h, and 2^t cells labelled by y, only containing object c_{n^3-t} (by considering $c_0 = c'$), object h_{n^3-t} (by considering $h_0 = h'$), object y_{n^3-t} (by considering $y_0 = y'$) respectively.*

(4) *There is a cell labelled by no, a cell labelled by yes and a cell labelled by out such that $\mathcal{C}_t(no) = \{w_{n^3-t+6}, no\}$, $\mathcal{C}_t(yes) = \{yes\}$, $\mathcal{C}_t(out) = \{x\}$.*

Proof: First of all, let us notice that at configuration $\mathcal{C}_{n\cdot p}$ no rule is applicable to any cell labelled by *in*, and no rule is applicable to cell labelled by *yes* or to cell labelled by *out*.

Now, let us show (2), (3) and (4) by induction on t. Let us start analyzing the basic case $t = n \cdot p + 1$.

(2) For each i, j, k $(1 \leq i, j, k \leq n)$, at configuration $\mathcal{C}_{n \cdot p}$ there are $2^{n \cdot p}$ cells labelled by $e_{i,j,k}$, each of them only containing object $(i,j)''_{k,n^3-n \cdot p}$. By applying a rule of the type (4) we will have $2^{n \cdot p+1}$ cells labelled by $e_{i,j,k}$ each of them containing $(i,j)''_{k,n^3-n \cdot p-1}$.

(3) For each i $(1 \leq i \leq n)$, at configuration $\mathcal{C}_{n \cdot p}$ there are $2^{n \cdot p}$ cells labelled by c_i, each of them only containing object $c_{n^3-n \cdot p}$. By applying a rule of the type (6) we will have $2 \cdot 2^{n \cdot p} = 2^{n \cdot p+1}$ cells labelled by c_i whose content is $c_{n^3-n \cdot p-1}$.
At configuration $\mathcal{C}_{n \cdot p}$, there are $2^{n \cdot p}$ cells labelled by h, each of them only containing object $h_{n^3-n \cdot p}$, and $2^{n \cdot p}$ cells labelled by y each of them only containing object $y_{n^3-n \cdot p}$. By applying a rule of type (8) we will have $2 \cdot 2^{n \cdot p} = 2^{n \cdot p+1}$ cells labelled by h whose content is $h_{n^3-n \cdot p-1}$. By applying a rule of type (7) we will have $2 \cdot 2^{n \cdot p} = 2^{n \cdot p+1}$ cells labelled by y whose content is $y_{n^3-n \cdot p-1}$.

(4) At configuration $\mathcal{C}_{n \cdot p}$, there is a cell labelled by *no* which contains object $w_{n^3-n \cdot p+6}$ and object *no*. By applying a rule of type (1), the new content of that cell will be objects $w_{n^3-n \cdot p-1+6}$ and *no*.

By induction hypothesis, let t be such that $n \cdot p+1 \leq t < n^3$ and let us suppose the result holds for t. Let us see that the result also holds for $t + 1$.

First of all, let us notice that at configuration $\mathcal{C}_t$ no rule is applicable to any cell labelled by *in*, and no rule is applicable to cell labelled by *yes* nor to cell labelled by *out* neither.

(2) For each i, j, k $(1 \leq i, j, k \leq n)$, according to induction hypothesis at configuration $\mathcal{C}_t$, there are 2^t cells labelled by $e_{i,j,k}$, and each of them only containing object $(i,j)''_{k,n^3-t}$. By applying a rule of type (4) we will have 2^{t+1} cells labelled by $e_{i,j,k}$ whose content is object $(i,j)''_{k,n^3-t-1}$.

(3) For each i $(1 \leq i \leq n)$, by induction hypothesis at configuration $\mathcal{C}_t$ there are 2^t cells labelled by c_i, each of them only containing object c_{n^3-t}. By applying a rule of the type (6) we will have $2 \cdot 2^t = 2^{t+1}$ cells labelled by c_i whose content is c_{n^3-t-1}.
By induction hypothesis, at configuration $\mathcal{C}_t$ there are 2^t cells labelled by h, each of them only containing object h_{n^3-t}, and 2^t cells labelled by y, each of them only containing object y_{n^3-t}. By applying a rule of type (8) we will have $2 \cdot 2^t = 2^{t+1}$ cells labelled by h whose content is object h_{n^3-t-1}. By applying a rule of the type (7) we will have $2 \cdot 2^t = 2^{t+1}$ cells labelled by y whose content is object y_{n^3-t-1}.

(4) By induction hypothesis, at configuration $\mathcal{C}_t$ there is a cell labelled by *no* which contains object w_{n^3-t+6} and object *no*. By applying a rule of the type (1) the new content of that cell is object $w_{n^3-t-1+6}$ and object *no*.

□

Corollary 5.4 *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ be a computation of the tissue P system $\Pi(n)$. Configuration $\mathcal{C}_{n^3}$ verifies the following:*

(1) *There are $2^{n \cdot p}$ cells labelled by in such that each of them contains a different subset of $A'_G = \{(u_i, v_i)'_k \mid 1 \leq i \leq p \wedge 1 \leq k \leq n \ \wedge \ (u_i, v_i) \in E\}$, as well as object y, object h and n copies of object c.*

(2) *For each i, j, k $(1 \leq i, j, k \leq n)$ there are 2^{n^3} cells labelled by $e_{i,j,k}$ each of them only containing object $(i, j)''_k$.*

(3) *For each i $(1 \leq i \leq n)$ there are 2^{n^3} cells labelled by c_i, 2^{n^3} cells labelled by h, and 2^{n^3} cells labelled by y, only containing object c', object h', object y' respectively.*

(4) *There is a cell labelled by no, a cell labelled by yes and a cell labelled by out such that $\mathcal{C}_{n^3}(no) = \{w_6, no\}$, $\mathcal{C}_{n^3}(yes) = \{yes\}$, $\mathcal{C}_{n^3}(out) = \{x\}$.*

Theorem 5.5 *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ be a computation of the tissue P system $\Pi(n)$. Configuration $\mathcal{C}_{n^3+1}$ verifies the following:*

(1) *There are $2^{n \cdot p}$ cells labelled by in such that each of them contains a different subset of $A''_G = \{(u_i, v_i)''_k \mid 1 \leq i \leq p \wedge 1 \leq k \leq n \ \wedge \ (u_i, v_i) \in E\}$, as well as object y, object h and n copies of object c.*

(2) *For each i, j, k $(1 \leq i, j, k \leq n)$ there are 2^{n^3} cells labelled by $e_{i,j,k}$ with the same content that at configuration $\mathcal{C}_{n^3}$ except whose cells $e_{v_i,v_j,k}$ in $\mathcal{C}_{n^3}$ which contains objects $(u_i, v_i)''_k \in A''_G$. At configuration $\mathcal{C}_{n^3+1}$ these objects are replaced by $(u_i, v_i)'_k$ respectively.*

(3) *For each i $(1 \leq i \leq n)$ (a) there are 2^{n^3} cells labelled by c_i from which $2^{n \cdot p}$ only contains object c, and the rest only contains object c', (b) there are 2^{n^3} cells labelled by h from which $2^{n \cdot p}$ only contains object h, and the rest only contains object h', and (c) there are 2^{n^3} cells labelled by y from which $2^{n \cdot p}$ only contains object y, and the rest only contains object y'.*

(4) *There is a cell labelled by no, a cell labelled by yes and a cell labelled by out such that $\mathcal{C}_{n^3+1}(no) = \{w_5, no\}$, $\mathcal{C}_{n^3+1}(yes) = \{yes\}$, $\mathcal{C}_{n^3+1}(out) = \{x\}$.*

Proof: It is enough to notice that configuration $\mathcal{C}_{n^3+1}$ is reached from $\mathcal{C}_{n^3}$ by applying:

- Rules of type (9) (from this follows (1) and (2)).
- Rules of type (10), (11), (12) and (1) (from this follows (3) and (4)).

1. Let us remark that at the $(n^3 + 1)$th step, we have replaced objects $(u_i, v_i)'_k$, with $1 \leq i \leq p \wedge 1 \leq k \leq n$, from cells labelled by in, by the respective objects $(u_i, v_i)''_k$ from cells labelled by $e_{u_i,v_i,k}$. Let us recall that at configuration $\mathcal{C}_{n^3}$ we had $2^{n \cdot p - 1}$ copies of objects $(u_i, v_i)'_k$, for each $1 \leq i \leq p \ \wedge \ 1 \leq k \leq n$ in cells labelled by in. Therefore, we need $2^{n \cdot p - 1}$ copies of objects $(u_i, v_i)''_k$ in cells labelled by $e_{u_i,v_i,k}$, but at configuration $\mathcal{C}_{n^3}$ we had 2^{n^3} copies of cells labelled by $e_{u_i,v_i,k}$, each of them only containing object $(u_i, v_i)''_k$.

2. Let us remark that at the (n^3+1)th step we have replaced m copies of object c in cell labelled by in by m respective copies of object c' from cells labelled by c_i $(1 \leq i \leq n)$. Then we need $n \cdot 2^{n \cdot p}$ copies of object c' in cells labelled by c_i. Let us recall that in total we had $n \cdot 2^{n^3}$ cells labelled by c_i, each of them only containing object c'. The same applies to objects h' and y' in cells labelled by h and y respectively.

□

Theorem 5.6 *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ be a computation of the tissue P system $\Pi(n)$. Configuration $\mathcal{C}_{n^3+2}$ verifies the following:*

(1) *There are $2^{n \cdot p}$ cells labelled by in such that each of them contains object y'', object h'', n copies of object c'', and a different subset of*

$$A''_G = \{(u_i, v_i)''_k \mid 1 \leq i \leq p \wedge 1 \leq k \leq n \ \wedge \ (u_i, v_i) \in E\}$$

of the form $\{(u_{\alpha_1}, v_{\alpha_1})''_{q_1}, (u_{\alpha_2}, v_{\alpha_2})''_{q_2}, \dots, (u_{\alpha_r}, v_{\alpha_r})''_{q_r}\}$, where

$$q_1 < q_2 < \ldots < q_r \ \wedge \ r \leq n \wedge (q_{i+1} = q_i + 1 \Rightarrow v_{\alpha_i} = u_{\alpha_{i+1}})$$

Moreover, each subset of A''_G verifying the previous conditions is contained inside some cell labelled by in.

(2) *Cells labelled by $e_{i,j,k}$, c_i, h, and y have the same content than at configuration $\mathcal{C}_{n^3+1}$.*

(3) *There is a cell labelled by no, a cell labelled by yes and a cell labelled by out such that $\mathcal{C}_{n^3+2}(no) = \{w_4, no\}$, $\mathcal{C}_{n^3+2}(yes) = \{yes\}$, $\mathcal{C}_{n^3+2}(out) = \{x\}$.*

Proof: It is enough to notice that configuration $\mathcal{C}_{n^3+1}$ yields $\mathcal{C}_{n^3+2}$ by applying rules of types (13), (14), (15), (16) and (1).

The first four rules are symport rules. Thus,at configuration $\mathcal{C}_{n^3+2}$ we will have $2^{n \cdot p}$ cells labelled by in in such manner that the subsets B of A''_G contained in each of them must verify the following conditions:

$(u, v)''_k \in B \wedge (u', v')''_{k'} \in B \wedge (u, v)''_k \neq (u', v')''_{k'} \Rightarrow u \neq u'$ (rule (13)).
$(u, v)''_k \in B \wedge (u', v')''_{k'} \in B \wedge (u, v)''_k \neq (u', v')''_{k'} \Rightarrow v \neq v'$ (rule (14)).
$(u, v)''_k \in B \wedge (u', v')''_{k'} \in B \wedge (u, v)''_k \neq (u', v')''_{k'} \Rightarrow k \neq k'$ (rule (16)).
$(u, v)''_k \in B \wedge (u', v')''_{k+1} \in B \Rightarrow v = u'$ (rule (15)).

Moreover, let us recall that at configuration $\mathcal{C}_{n^3+1}$, every subset B of A''_G is contained inside a different cell labelled by in. Therefore, each subset B of A''_G verifying the previous conditions will be contained inside one unique cell labelled by in at configuration $\mathcal{C}_{n^3+2}$.

□

Remark: From Proposition 4.2, we deduce that a subset B from A''_G represents a Hamiltonian cycle of G if and only if $|B| = n$ and B satisfies the conditions $(\alpha), (\beta), (\gamma), (\delta)$. Thus, to determine whether or not graph G has a Hamiltonian cycle will be equivalent to determine whether or not in some cell of configuration $\mathcal{C}_{n^3+2}$ labelled by in there exists a subset B from A''_G whose cardinality is n.

Theorem 5.7 *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ be a computation of the tissue P system $\Pi(n)$. Configuration $\mathcal{C}_{n^3+3}$ verifies the following:*

(1) *There are $2^{n \cdot p}$ cells labelled by in such that each of them contains a copy of object y''' and a copy of object h'''. Besides, if a subset $B \subseteq A''_G = \{(u_i, v_i)''_k \mid 1 \leq i \leq p \wedge 1 \leq k \leq n \wedge (u_i, v_i) \in E\}$ contained in a cell labelled by in has exactly n elements, then no object c'' appears inside that cell. Otherwise, inside any cell labelled by in which contains a subset $B \subseteq A''_G$, some objects c'' (exactly $n - t_1$ copies, where t_1 is the size of the subset B) will remain.*

(2) *Cells labelled by $e_{i,j,k}$, c_i, h, and y have the same content than at configuration $\mathcal{C}_{n^3+1}$.*

(3) *There is a cell labelled by no, a cell labelled by yes and a cell labelled by out such that $\mathcal{C}_{n^3+3}(no) = \{w_3, no\}$, $\mathcal{C}_{n^3+3}(yes) = \{yes\}$, $\mathcal{C}_{n^3+3}(out) = \{x\}$.*

Proof: It is enough to notice that configuration $\mathcal{C}_{n^3+2}$ yields $\mathcal{C}_{n^3+3}$ by applying rules of types (20), (21), (22), and (1).

By applying rules of types (21) and (22), objects h'' and y'' evolve to h''' and y''' respectively.

By applying rules of type (20), for each element $(u, v)''_k$ in the set encoded by that cell, one object c'' will be consumed.

By applying a rule of type (1), object w_4 evolves to object w_3.

□

Corollary 5.8 *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ be a computation of the tissue P system $\Pi(n)$. The following assertions are equivalent:*

- *Graph G has a Hamiltonian cycle.*
- *At configuration $\mathcal{C}_{n^3+3}$, there is, at least, a cell labelled by in such that it does not contain any object c''.*

Proof: It suffices to notice that, at configuration $\mathcal{C}_{n^3+2}$, Hamiltonian cycles are characterized by membranes labelled by in which contain a subset of A''_G of size n. Then, by using a rule of type (20), at configuration $\mathcal{C}_{n^3+3}$ Hamiltonian cycles are characterized by membranes labelled by in such that they do not contain any object c''.

Theorem 5.9 *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ be a computation of the tissue P system $\Pi(n)$. Configuration $\mathcal{C}_{n^3+4}$ verifies the following properties:*

(1) *There are $2^{n \cdot p}$ cells labelled by in such that each of them contains one copy of objects y''''. Besides, if object c'' appeared in some cell labelled by in at configuration $\mathcal{C}_{n^3+3}$, then that copy of c'' and object h''' are released out to the environment. Otherwise, object h''' will remain inside that cell in at configuration $\mathcal{C}_{n^3+4}$.*

(2) *Cells labelled by $e_{i,j,k}$, c_i, h, and y have the same content than at configuration $\mathcal{C}_{n^3+1}$.*

(3) *There is a cell labelled by no, a cell labelled by yes and a cell labelled by out such that* $\mathcal{C}_{n^3+4}(no) = \{w_2, no\}$, $\mathcal{C}_{n^3+4}(yes) = \{yes\}$, $\mathcal{C}_{n^3+4}(out) = \{x\}$.

Proof: It is enough to notice that configuration $\mathcal{C}_{n^3+3}$ yields $\mathcal{C}_{n^3+4}$ by applying rules of types (23), (24), and (1).

By applying rules of type (23), object h'' and a copy of object c'' (if any in a cell labelled by *in*) will be released to the environment.

By applying a rule of type (24), object y''' evolves to object y''''.

By applying a rule of type (1), object w_3 evolves to object w_2.

□

Corollary 5.10 *Let* $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ *be a computation of the tissue P system* $\Pi(n)$. *The following assertions are equivalent:*

- *Graph G has a Hamiltonian cycle.*
- *At configuration* $\mathcal{C}_{n^3+4}$, *there is, at least, a cell labelled by in at configuration* $\mathcal{C}_{n^3+4}$ *such that it contains an object* h'''.

Besides, graph G has exactly q Hamiltonian cycles if and only if there are exactly $n \cdot q$ *cells labelled by in such that at configuration* $\mathcal{C}_{n^3+4}$, *it contains an object* h'''.

Proof: It suffices to notice that, at configuration $\mathcal{C}_{n^3+3}$, Hamiltonian cycles are characterized by membranes labelled by *in* such that they do not contain any object c''. Then, a rule of type (23) will be applicable to each cell labelled by *in* that contains some object c''. In this case, at configuration $\mathcal{C}_{n^3+4}$, object h''' will only appear at membranes labelled by *in* which encode Hamiltonian cycles.

Theorem 5.11 *Let* $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ *be a computation of the tissue P system* $\Pi(n)$. *Configuration* $\mathcal{C}_{n^3+5}$ *verifies the following:*

(1) *There are* $2^{n \cdot p}$ *cells labelled by in such that each of them contains one copy of objects* y'''. *Besides, if object* h''' *appeared in some cell labelled by in at configuration* $\mathcal{C}_{n^3+4}$, *then object* h''' *together with object* y''' *are sent to cell labelled by yes. Otherwise, that cell in remain unchanged at the next configuration.*

(2) *Cells labelled by* $e_{i,j,k}$, c_i, h, *and* y *have the same content than at configuration* $\mathcal{C}_{n^3+1}$.

(3) *There is a cell labelled by no such that:* $\mathcal{C}_{n^3+5}(no) = \{w_1, no\}$.

(4) *There is a cell labelled by yes which contains either only object yes, or* $n \cdot q$ *copies of objects* y'''' *and* $n \cdot q$ *copies of objects* h''', *and object yes, being q the total number of Hamiltonian cycle of G. Besides, there is a cell labelled by out which contains only object x.*

Proof: It is enough to notice that configuration $\mathcal{C}_{n^3+4}$ yields $\mathcal{C}_{n^3+5}$ by applying rules of types (23), (24), and (1).

By applying rules of type (25) to each cell labelled by *in* that encodes a Hamiltonian cycle, object h''' and a copy of object c'' will be released to the environment. If G has exactly q Hamiltonan cycles, then there are exactly $n \cdot q$ cells labelled by

in that encodes a Hamiltonian cycle. In this case, the contents of cell labelled by *yes* will be $n \cdot k$ copies of object y'''', $n \cdot k$ copies of object h''' and object *yes*.

By applying a rule of type (1), object w_2 evolves to object w_1 in cell labelled by *no*.

□

Corollary 5.12 *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ be a computation of the tissue P system $\Pi(n)$. The following assertions are equivalent:*

- *Graph G has a Hamiltonian cycle.*
- *At configuration $\mathcal{C}_{n^3+5}$, cell labelled by yes has, at least, a copy of object y'''' and a copy of object yes.*

Besides, graph G has exactly q Hamiltonian cycles if and only if at configuration $\mathcal{C}_{n^3+5}$, the cell labelled by yes has exactly $n \cdot q$ copies of objects y''' and a copy of object yes.

Proof: It suffices to notice that, at configuration $\mathcal{C}_{n^3+4}$, Hamiltonian cycles are characterized by membranes labelled by *in* which contain object h'''. Then, a rule of type (25) will be applicable to these cell, producing object h'''' in the cell labelled by *yes*.

Theorem 5.13 *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ be a computation of the tissue P system $\Pi(n)$. Configuration $\mathcal{C}_{n^3+6}$ verifies the following:*

(1) *Cells labellled by in, $e_{i,j,k}$, c_i, h, and y have the same content than at configuration $\mathcal{C}_{n^3+5}$.*
(2) *There is a cell labelled by no which contains object w and object no.*
(3) *There is a cell labelled by yes which contains either only object yes (in this case there is a cell labelled by out which contains only object x), or contains $n \cdot q - 1$ copies of objects y'''' and $n \cdot q$ copies of objects h''', being q the total number of Hamiltonian cycles of G (in this case, there is a cell labelled by out which contains object y'''', object yes and object x).*

Proof: It is enough to notice that configuration $\mathcal{C}_{n^3+5}$ yields $\mathcal{C}_{n^3+6}$ by applying rules of types (26) and (2).

If there is, at least, an object y'''' in cell labelled by *yes*, then by applying rule (26), a copy of object y'''' and object *yes* are sent to the cell labelled by *out*. Otherwise, that rule is not applicable to cell *yes*.

In any case, by applying rule (2) to cell labelled by *no*, object w_1 evolves to object w.

□

Corollary 5.14 *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ be a computation of the tissue P system $\Pi(n)$. The following assertions are equivalent:*

- *Graph G has a Hamiltonian cycle.*

- *At configuration $\mathcal{C}_{n^3+6}$, cell labelled by out has a copy of object yes and a copy of object x.*

Proof: It suffices to notice that, at configuration $\mathcal{C}_{n^3+5}$, Hamiltonian cycles are characterized by the following condition: cell labelled by *yes* contains objects h'''' and *yes*. Then, a rule of type (26) is applicable to cell labelled by *yes* sending these objects to the celll labelled by *out*.

Theorem 5.15 *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ be a computation of the tissue P system $\Pi(n)$. Configuration $\mathcal{C}_{n^3+7}$ verifies the following properties:*

(1) *Cells labelled by in, $e_{i,j,k}$, c_i, h, y, and yes have the same content than at configuration $\mathcal{C}_{n^3+6}$.*
(2) *If G has a Hamiltonian cycle, then $\mathcal{C}_{n^3+7}(no) = \emptyset$, $\mathcal{C}_{n^3+7}(yes) = \{h'''\}$, $\mathcal{C}_{n^3+7}(out) = \{y''''', w, no\}$, and $yes, x \in \mathcal{C}_{n^3+7}(0)$. The configuration $\mathcal{C}_{n^3+7}$ is a halting configuration. Moreover, it is an accepting configuration.*
(3) *If G doesn't have a Hamiltonian cycle, then $\mathcal{C}_{n^3+7}(no) = \emptyset$, $\mathcal{C}_{n^3+7}(yes) = \{yes\}$, $\mathcal{C}_{n^3+7}(out) = \{w, no, x\}$.*

Proof: It is enough to notice that configuration $\mathcal{C}_{n^3+6}$ yields $\mathcal{C}_{n^3+7}$ by applying rules of the type (27), in the case that G has a Hamiltonian cycle, and (28) in any case. Besides, by applying rule of type (27) objects x and *yes* are sent to the environment. Thus, $x \notin \mathcal{C}_{n^3+7}(out)$ and the rule of type (29) will not be applicable at the next step. Hence, configuration $\mathcal{C}_{n^3+7}$ is an accepting configuration.

□

Theorem 5.16 *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots)$ be a computation of the tissue P system $\Pi(n)$. Let us suppose that G doesn't have a Hamiltonian cycle, then configuration $\mathcal{C}_{n^3+8}$ verifies the following:*

(1) *Cells labelled by in, $e_{i,j,k}$, c_i, h, y, no and yes, have the same content than at configuration $\mathcal{C}_{n^3+7}$.*
(2) *$\mathcal{C}_{n^3+8}(no) = \emptyset$, $\mathcal{C}_{n^3+8}(yes) = \{yes\}$, $\mathcal{C}_{n^3+8}(out) = \{w\}$, and $no, x \in \mathcal{C}_{n^3+8}(0)$. The configuration $\mathcal{C}_{n^3+8}$ is a halting configuration. Moreover, it is a rejecting configuration.*

Proof: It is enough to notice that if G doesn't have a Hamiltonian cycle, then configuration $\mathcal{C}_{n^3+8}$ is reached from $\mathcal{C}_{n^3+7}$ by applying the rule of type (29).

□

Corollary 5.17 *The family $\mathbf{\Pi}$ defined at Section 4.2 is polynomially bounded with regard to* $(\texttt{HAM-CYCLE}, cod, s)$.

Proof: From Theorem 5.15 and Theorem 5.16, we deduce that any computation $\mathcal{C}$ of the tissue P system $\Pi(n)$ spends $n^3 + 7$ or $n^3 + 8$ transition steps, for each $n \in \mathbb{N}$.

□

Corollary 5.18 *The family* $\mathbf{\Pi}$ *defined at Section 4.2 is sound and complete with regard to* (HAM-CYCLE, cod, s)

Proof: Let G be a directed graph that has a Hamiltonian cycle. Let $\mathcal{C}$ be an arbitrary computation of $\Pi(s(G)) + cod(G)$. From Theorem 5.15, we deduce that $\mathcal{C}$ is an accepting computation.

Now, let G be a directed graph such that there exists an accepting computation $\mathcal{C}$ of $\Pi(s(G)) + cod(G)$. Then, G has a Hamiltonian cycle. Otherwise, computation $\mathcal{C}$ must be a rejecting computation according to Theorem 5.16.

□

Theorem 5.19 HAM-CYCLE $\in \mathbf{PMC}_{TDC(2)}$.

Proof: The family of tissue P systems with cell division constructed in Subsection 4.2 verifies the following:

(a) Every system of the family $\mathbf{\Pi}$ is a recognizer tissue P system with cell division and communication rules with length at most 2.
(b) The family $\mathbf{\Pi}$ is polynomially uniform by Turing machines (Subsection 5.1).
(c) The pair (cod, s) of polynomial–time computable functions defined in Subsection 4.3 verifies: for each instance G of HAM-CYCLE, $s(G)$ is a natural number, $cod(G)$ is an input multiset of the system $\Pi(s(G))$, and for each $n \in \mathbb{N}$, $s^{-1}(n)$ is a finite set.
(d) The family $\mathbf{\Pi}$ is polynomially bounded with regard to (HAM-CYCLE, cod, s) (Corollary 5.17).
(e) The family $\mathbf{\Pi}$ is sound and complete with regard to (HAM-CYCLE, cod, s) (Corollary 5.18).

Therefore, according to Definition 2.4, the uniform family $\mathbf{\Pi}$ of tissue P systems constructed in Section 4 solves the HAM-CYCLE problem in polynomial time with respect to the number of variables and the number of clauses.

□

Corollary 5.20 $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{TDC(2)}$.

Proof: It suffices to notice that the HAM-CYCLE problem is **NP**-complete, HAM-CYCLE$\in \mathbf{PMC}_{TDC(2)}$, and this complexity class is closed under polynomial-time reduction and under complement.

□

6 Conclusions

The length of communication rules plays a relevant role for tissue P systems with cell division from the efficiency point of view. A uniform and efficient solution to the Vertex Cover problem by using a family of tissue P systems with cell division and communication rules of length at most 3 was given in [1]. By using the dependency

graph technique of cell–like P systems, it was shown that only tractable problems can be efficiently solved by using families of tissue P systems with cell division and communication rules of length 1 [3]. Hence, assuming that $\mathbf{P} \neq \mathbf{NP}$, in the framework of tissue P systems with cell division, passing from communication rules of length 1 to communication rules of length at most 3 amounts to passing from non–efficiency to efficiency.

In this paper, that borderline of efficiency has been optimized by proving that a well known **NP**–complete problem, the `HAM-CYCLE` problem, can be solved in a uniform and efficient way, by using a family of tissue P systems with cell division and communication rules of length at most 2.

In [7], cell separation rules were introduced into tissue P systems (inspired by the cellular fission) and their computational efficiency was investigated. Two important results were obtained in that framework: (a) only tractable problems can be efficiently solved by using cell separation and communication rules with length at most 1, and (b) a uniform and linear time solution to the `SAT` problem by using cell separation and communication rules with length at most 8 was presented. Recently [14] this result was improved by showing a family of tissue P systems with cell separation and communication rules with length at most 3, solving the `SAT` problem in a uniform way and linear time.

Now, we propose three open problems related to the efficiency of tissue P systems:

(a) What is the computational efficiency of tissue P systems with cell separation which allow communication rules with length at most 2?
(b) What happens if only symport (or only antiport) rules are allowed in tissue P systems with cell division or cell separation?
(c) At the initial configuration of a tissue P system the symbols of the alphabet $\mathcal{E}$ appear in the environment in an *arbitrary number of copies*. We can consider tissue P systems without environment, that is, tissue P systems where alphabet $\mathcal{E}$ is empty. What is the relationship between the polynomial complexity classes of tissue P systems with cell division (or with cell separation) and the corresponding tissue P systems without environment?

Acknowledgements

The work was supported by TIN2009-13192 Project 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. Antonio E. Porreca was partially supported by Università degli Studi di Milano-Bicocca, Fondo di Ateneo per la Ricerca (FAR) 2011. Research by Niall Murphy was partly supported by a PICATA postdoctoral fellowship of the Moncloa Campus of International Excellence (UCM-UPM)

References

1. D. Díaz-Pernil, M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Núñez, F.J. Romero–Campero. Computational efficiency of cellular division in tissue-like P systems. *Romanian Journal of Information Science and Technology* **11**, 3, (2008), 229–241.
2. M.R. Garey, D.S. Johnson. *Computers and Intractability A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, (1979).
3. R. Gutiérrez-Escudero, M.J. Pérez-Jiménez, M. Rius–Font. Characterizing tractability by tissue-like P systems. *Lecture Notes in Computer Science* **5957**, (2010), 289–300.
4. M. Ito, C. Martín Vide, Gh. Păun. A characterization of Parikh sets of ET0L laguages in terms of P systems. In M. Ito, Gh. Păun, S. Yu (eds.) *Words, Semigroups and Transducers*, World Scientific, Singapore, 2001, 239-254.
5. C. Martín Vide, J. Pazos, Gh. Păun, A. Rodríguez Patón. A New Class of Symbolic Abstract Neural Nets: Tissue P Systems. *Lecture Notes in Computer Science* **2387**, (2002), 290–299.
6. L. Pan, T.-O. Ishdorj. P systems with active membranes and separation rules. *Journal of Universal Computer Science*, **10**, 5, (2004), 630–649.
7. L. Pan, M.J. Pérez-Jiménez. Computational complexity of tissue–like P systems. *Journal of Complexity*, **26**, 3 (2010), 296–315.
8. Gh. Păun. Attacking **NP**-complete problems. In *Unconventional Models of Computation, UMC'2K* (I. Antoniou, C. Calude, M. J. Dinneen, eds.), Springer-Verlag, 2000, pp. 94-115.
9. Gh. Păun. *Membrane Computing. An Introduction*. Springer–Verlag, Berlin, (2002).
10. A. Păun, Gh. Păun. The power of communication: P systems with symport/antiport. *New Generation Computing*, **20**, 3, (2002), 295–305.
11. Gh. Păun, M.J. Pérez-Jiménez, A. Riscos-Núñez. Tissue P System with cell division. *In. J. of Computers, Communications and Control*, **3**, 3, (2008), 295–303.
12. M.J. Pérez-Jiménez, A. Romero-Jiménez, Sancho-Caparrini. Complexity classes in models of cellular computing with membranes. *Natural Computing*, **2**, 3 (2003), 265–285.
13. M.J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini. A polynomial complexity class in P systems using membrane division. *Journal of Automata, Languages and Combinatorics*, **11**, 4, (2006), 423-434.
14. M.J. Pérez-Jiménez, P. Sosík. Improving the efficiency of tissue P systems with cell separation. Submitted, 2012.
15. H. Vollmer. *Introduction to Circuit Complexity: A Uniform Approach*. Springer–Verlag, New York, (1999).