A uniform family of tissue P systems with cell division solving 3-COL in a linear time

Daniel Díaz-Pernil, Miguel A. Gutiérrez-Naranjo, Mario J. Pérez-Jiménez *,
Agustín Riscos-Núñez

Research Group on Natural Computing, Universidad de Sevilla, Sevilla, Spain

A B S T R A C T

Several examples of the efficiency of cell-like P systems regarding the solution of **NP**-complete problems in polynomial time can be found in the literature(obviously, trading space for time). Recently, different new models of tissue-like P systems have received much attention from the scientific community. In this paper we present a linear-time solution to an **NP**-complete problem from graph theory, the 3-coloring problem, and we discuss the suitability of tissue-like P systems as a framework to address the efficient solution to intractable problems.

Keywords:
Membrane computing
Tissue P systems
Cell division
3-coloring problem

1. Introduction

This paper is within the Natural Computing framework and, more precisely, in the study of the structure and functioning of cells as living organisms able to process and generate information. We focus on membranes, which are involved in many reactions taking place inside various compartments of a cell, and they act as selective channels of communication between different compartments as well as between the cell and its environment [1]. Assuming this starting point, two different disciplines within Natural Computing can be found in the literature: *Membrane Computing* and *Brane Calculi*.

Brane Calculi were recently introduced in [6], under the assumption that in living cells membranes are not merely containers, but they are actually highly dynamic and participate actively in the cell life. In this way, "computation" happens on the membranes, and not inside them.

On the other hand, Membrane Computing starts from the assumption that the processes taking place within the compartmental structure of a living cell can be interpreted as computations [18].

This emergent cross-disciplinary branch of Natural Computing was introduced by Păun in [17]. It has received important attention from the scientific community since then, with contributions from computer scientists, biologists, formal linguists and complexity theoreticians, enriching each other with results, open problems and promising new research lines. In fact, Membrane Computing was selected by the Institute for Scientific Information, USA, as a fast *Emerging Research Front* in Computer Science, and [19] was mentioned in [29] as a highly cited paper in October 2003.

The computational devices in Membrane Computing are called *P systems*. Roughly speaking, a P system consists of a membrane structure, in the compartments of which one places multisets of objects which evolve according to given rules in a synchronous non-deterministic maximally parallel manner.[1]

In recent years, many different models of P systems have been proposed. The most studied variants are characterized by a *cell-like* membrane structure, where the communication happens between a membrane and the surrounding one. In this model we have a set of nested membranes, in such a way that the graph of neighborhood relation is a tree.

* Corresponding author.
E-mail address: marper@us.es (M.J. Pérez-Jiménez).

[1] A layman-oriented introduction can be found in [20] and further bibliography at [30].

One of the topics in the field is the study of the computational power and efficiency of P systems. In particular, different models of these cell-like P systems have been successfully used in order to design solutions to **NP**-complete problems in polynomial time (see [9] and the references therein). These solutions are obtained by generating an exponential amount of workspace in polynomial time and using parallelism to check simultaneously all the candidate solutions. Inspired by living cells, cell-like P systems abstract the way of obtaining new membranes, mainly from two biological processes: *mitosis* (membrane division) and *autopoiesis*, see [13] (membrane creation). Both ways of generating new membranes have given rise to the corresponding P systems model: *P systems with active membranes*, where the new workspace is generated by membrane division and *P systems with membrane creation*, where the new membranes are created from objects.

Both models are universal from a computational point of view, but technically, they are quite different. In fact, there are no theoretical results which prove that these models can simulate each other in polynomial time.

Under the hypothesis $\mathbf{P} \neq \mathbf{NP}$, Zandron et al. [28] established the limitations of P systems that do not use membrane division concerning the efficient solution of **NP**-complete problems. This result was generalized by Pérez-Jiménez et al. [24] obtaining a characterization of the $\mathbf{P} \neq \mathbf{NP}$ conjecture by the polynomial time unsolvability of an **NP**-complete problem by language accepting P systems (without using rules that allow to construct an exponential number of membranes in polynomial time).

We shall focus here on another type of P systems, the so-called *Tissue P Systems*(because of their membrane structure). Instead of considering a hierarchical arrangement, membranes are placed in the nodes of a graph. This variant has two biological inspirations (see [16]): 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. The communication among cells is based on symport/antiport rules.[2] Symport rules move objects across a membrane together in one direction, whereas antiport rules move objects across a membrane in opposite directions.

From the seminal definition of Tissue P systems [15,16], several research lines have been developed and other variants have arisen (see, for example, [2,5,7,11,12,26]). One of the most interesting variants of Tissue P systems was presented in [21]. In that paper, the definition of Tissue P systems is combined with the one of P systems with active membranes, yielding *Tissue P systems with cell division*.

One of the main features of such Tissue P systems with cell division is related to their computational efficiency. In [21], a polynomial-time solution to the **NP**-complete problem SAT is shown. In this paper we go on with the research in this model and present a linear-time solution to another well-known **NP**-complete problem: the 3-coloring problem.

The paper is organized as follows: first we recall some preliminaries and the definition of Tissue P systems with cell division. Next, recognizer Tissue P systems are briefly described. A linear–time solution to the 3-coloring problem is presented in the following section, including a short overview of the computation and of the necessary resources. We also include the formal verification of the solution. Finally, the main results, some conclusions and new open research lines are presented.

2. Preliminaries

In this section we briefly recall some of the concepts used later on in the paper.

An *alphabet*, Σ, is a non-empty set, whose elements are called *symbols*. An ordered sequence of symbols is a *string*. 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 strings of length n built with symbols from the alphabet Σ is denoted by Σ^n and $\Sigma^* = \cup_{n\geq 0}\Sigma^n$. A *language* over Σ is a subset from Σ^*.

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 $supp(m) = \{x \in A \,|\, f(x) > 0\}$ and its *size* is defined as $\sum_{x\in A} f(x)$. A multiset is empty (resp. finite) if its support is the empty set (resp. finite).

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 any $x \in A$, then this element is omitted.

A *graph* G is a pair $G = (V, E)$ where V is the set of vertices and E is the set of edges, each one of which is a (unordered) pair of (different) vertices. If $\{u, v\} \in E$, we say that u is *adjacent* to v (and also v is *adjacent* to u). The *degree* of $v \in V$ is the number of adjacent vertices to v.

In what follows we assume the reader is already familiar with the basic notions and the terminology underlying P systems. For details, see [18].

3. Tissue P systems with cell division

In the first definition of the model of tissue P systems [15,16] the membrane structure did not change along the computation. Based on the cell-like model of P systems with active membranes, Păun et al. presented in [21] a new model

[2] This way of communication for P systems was introduced in [19].

of tissue P systems *with cell division*. The biological inspiration is clear: alive tissues are not *static* network of cells, since cells are duplicated via mitosis in a natural way.

The main features of this model, from the computational point of view, are that cells are not polarized (the contrary holds in the cell-like model of P systems with active membranes, see [18]); 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 blocked during the mitosis process. In some sense, this means that while a cell is dividing, it closes its communication channels.

Formally, a *tissue P system with cell division* of degree $q \geq 1$ is a tuple of the form

$$\Pi = (\Gamma, w_1, \ldots, w_q, \mathcal{E}, \mathcal{R}, i_0),$$

where:

(1) Γ is a finite *alphabet*, whose symbols will be called *objects*.
(2) $w_1, \ldots, w_q$ are strings over Γ.
(3) $\mathcal{E} \subseteq \Gamma$ is the alphabet of the environment.
(4) $\mathcal{R}$ is a finite set of rules of the following form:
  (a) *Communication rules*: $(i, u/v, j)$, for $i, j \in \{0, 1, 2, \ldots, q\}$, $i \neq j$, $u, v \in \Gamma^*$.
  (b) *Division rules*: $[a]_i \to [b]_i[c]_i$, where $i \in \{1, 2, \ldots, q\}$ and $a, b, c \in \Gamma$.
(5) $i_0 \in \{0, 1, 2, \ldots, q\}$.

A tissue P system with cell division of degree $q \geq 1$ can be seen as a set of q cells (each one consisting of an elementary membrane) labelled by $1, 2, \ldots, q$. We shall use 0 as the label of the environment, and i_0 denotes the output region (which can be a region inside a membrane or the environment).

The communication rules determine an implicit net of channels, where the nodes are the cells and the edges indicate if it is possible for pairs of cells to communicate directly. This is a dynamical graph, as new nodes can appear produced by the application of division rules. Note also that the connections depend only on the label of the cell, and thus when a cell is divided, the two new cells will have identical connections. Nevertheless, this graph is just an intuition, we shall not handle it explicitly along the computations.

The strings $w_1, \ldots, w_q$ describe the multisets of objects placed in the q cells of the system. We interpret that $\mathcal{E} \subseteq \Gamma$ is the set of objects placed in the environment, each one of them in an arbitrary large amount of copies.

The communication rule $(i, u/v, j)$ can be applied over two cells i and j such that u is contained in cell i and v is contained in cell j. The application of this rule means that the objects of the multisets represented by u and v are interchanged between the two cells.

The division rule $[a]_i \to [b]_i[c]_i$ can be applied over a cell i containing object a. The application of this rule divides this cell into two new cells with the same label. All the objects in the original cell are replicated and copied in each of the new cells, with the exception of the object a, which is replaced by the object b in the first new cell and by c in the second one.

Rules are used as usual in the framework of membrane computing, that is, in a maximally parallel way. In one step, each object in a membrane can only be used for one rule (non-deterministically chosen when there are several possibilities), but any object which can participate in a rule of any form must do it, i.e in each step we apply a maximal set of rules. This way of applying rules has only one restriction when a cell is divided, the division rule is the only one which is applied for that cell in that step; the objects inside that cell do not evolve in that step.

4. Recognizer tissue P systems with cell division

NP-completeness has been usually studied in the framework of *decision problems*. 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.

In order to study the computing efficiency for solving **NP**-complete decision problems, a special class of tissue P systems with cell division is introduced in [21]: *recognizer tissue P systems*. The key idea of such recognizer system is the same one as from recognizer P systems with cell-like structure.

Recognizer cell-like P systems were introduced in [25] and they are the natural framework to study and solve decision problems within Membrane Computing, since deciding whether an instance has an affirmative or negative answer is equivalent to deciding whether or not a string belongs to the language associated with the problem.

In the literature, recognizer cell-like P systems are associated in a natural way with P systems with *input*. The data related to an instance of the decision problem has to be provided to the P system in order to compute the appropriate answer. This is done by encoding each instance as a multiset placed in an *input membrane*. The output of the computation (`yes` or `no`) is sent to the environment. In this way, cell-like P systems with input and external output are devices which can be seen as black boxes, in the sense that the user provides the data before the computation starts, and then waits *outside* the P system until it sends to the environment the output in the last step of the computation (see [23] for details).

A recognizer tissue P system with cell division of degree $q \geq 1$ is a tuple

$$\Pi = (\Gamma, \Sigma, w_1, \ldots, w_q, \mathcal{E}, \mathcal{R}, i_{in}, i_0)$$

where

- $(\Gamma, w_1, \ldots, w_q, \mathcal{E}, \mathcal{R}, i_o)$ is a tissue P system with cell division of degree $q \geq 1$ (as defined in the previous section).
- The working alphabet Γ has two distinguished objects $\mathtt{yes}$ and $\mathtt{no}$, present in at least one copy in some initial multisets $w_1, \ldots, w_q$, but not present in $\mathcal{E}$.
- Σ is an (input) alphabet strictly contained in Γ.
- $i_{in} \in \{1, \ldots, q\}$ is the input cell.
- The output region i_0 is the environment.
- All computations halt.
- If $\mathcal{C}$ is a computation of Π, then either the object $\mathtt{yes}$ or the object $\mathtt{no}$ (but not both) must have been released into the environment, and only in the last step of the computation.

The computations of the system Π with input $w \in \Sigma^*$ start from a configuration of the form $(w_1, w_2, \ldots, w_{i_{in}}w, \ldots, w_q; \mathcal{E})$, that is, after adding the multiset w to the contents of the input cell i_{in}. We say that the multiset w is *recognized* by Π if and only if the object $\mathtt{yes}$ is sent to the environment, in the last step of the corresponding computation. We say that $\mathcal{C}$ is an accepting computation or rejecting computation if the object $\mathtt{yes}$ or $\mathtt{no}$), respectively, appears in the environment associated with the corresponding halting configuration of $\mathcal{C}$.

Definition 1. We say that a decision problem $X = (I_X, \theta_X)$ is solvable in polynomial time by a family $\mathbf{\Pi} = \{\Pi(n) : n \in \mathbb{N}\}$ of recognizer tissue P systems with cell division if the following holds:

- The family $\mathbf{\Pi}$ is *polynomially uniform* by Turing machines, that is, there exists a deterministic Turing machine working in polynomial time which constructs the system $\Pi(n)$ from $n \in \mathbb{N}$.
- There exists a pair (cod, s) of polynomial-time computable functions over I_X such that:
  - for each instance $u \in I_X$, $s(u)$ is a natural number and $cod(u)$ is an input multiset of the system $\Pi(s(u))$;
  - 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, moreover, it performs at most $p(|u|)$ steps;
  - the family $\mathbf{\Pi}$ is *sound* with regard to (X, cod, s), that is, for each $u \in I_X$, if there exists an accepting computation of $\Pi(s(u))$ with input $cod(u)$, then $\theta_X(u) = 1$;
  - the family $\mathbf{\Pi}$ is *complete* with regard to (X, cod, s), that is, for each $u \in I_X$, if $\theta_X(u) = 1$, then every computation of $\Pi(s(u))$ with input $cod(u)$ is an accepting one.

In the above definition we assume every P system $\Pi(n)$ to be *confluent*, in the following sense: every computation of a system with the *same* input multiset must always give the *same* answer.

We denote by $\mathbf{PMC}_{TD}$ the set of all decision problems which can be solved by means of recognizer tissue P systems with cell division in polynomial time.

5. A solution for the 3-coloring problem

A k-coloring ($k \geq 1$) of an undirected graph $\mathcal{G} = (V, E)$ is a function $f : V \to \{1, \ldots, k\}$, where the numbers are interpreted as colors. We say that $\mathcal{G}$ is k-colorable if there exists a k-coloring, f, such that $f(u) \neq f(v)$ for every edge $\{u, v\} \in E$ (such a k-coloring f is said to be *valid*).

The 3-coloring problem is the following: *given an undirected graph $\mathcal{G}$, decide whether or not $\mathcal{G}$ is* 3-*colorable*; that is, if there exists a valid 3-coloring of $\mathcal{G}$. For the sake of readability, we shall use $\{R, G, B\}$ instead of $\{1, 2, 3\}$ to represent the colors (R, G and B standing for *red, green* and *blue*, respectively).

This problem is related to the famous Four Color Conjecture (proved by Appel and Haken [3,4]). It is a particular case of the colorability problem: *Given an undirected graph $\mathcal{G}$ and a number k, decide whether $\mathcal{G}$ is k-colorable*. The **NP**-completeness of the 3-coloring problem was proved by Stockmeyer [27] (see [8]).

Next, we shall prove that the 3-coloring problem can be solved in a linear time by a family of recognizer tissue P systems with cell division. We shall address the resolution via a brute force algorithm, which consists in the following stages:

- *Generation stage*: The initial cell labelled by 2 is divided into two new cells; and the divisions are iterated until a cell has been produced for each possible candidate solution. Simultaneously, in the cell labelled by 1 there is a counter that will determine the moment in which the checking stage starts.
- *Pre–checking stage*: After obtaining all possible 3-colorings encoded in cells labelled by 2, this stage provides objects R_{ij}, G_{ij}, B_{ij} in such cells, for every edge A_{ij}.
- *Checking stage*: Objects R_{ij}, G_{ij}, B_{ij} will be used in cells labelled by 2 to check if there exists a pair of adjacent vertices with the same color in the corresponding candidate solution (R_{ij} checks if nodes v_i and v_j are both *red*, and analogously G_{ij} and B_{ij} for *green* and *blue* colors).
- *Output stage*: The system sends to the environment the right answer according to the results of the previous stage.

Let us recall that the function $\langle n, m\rangle = ((n+m)(n+m+1)/2) + n$ is a primitive recursive bijection between $\mathbb{N}^2$ and $\mathbb{N}$. Moreover, its inverse function is also polynomial. Next, we shall define a family $\mathbf{\Pi} = \{\Pi(i) : i \in \mathbb{N}\}$ such that each system $\Pi(i)$ will solve all instances of graphs with n vertices and m edges, where $i = \langle n, m\rangle$, provided that the appropriate input multiset is provided.

For each $n, m \in \mathbb{N}$, we shall consider the system

$$\Pi(\langle n, m\rangle) = (\Gamma(\langle n, m\rangle), \Sigma(n), w_1, w_2(n), \mathcal{R}(\langle n, m\rangle), \mathcal{E}(\langle n, m\rangle), i_{in}, i_0)$$

where:

- $\Gamma(\langle n, m\rangle)$ is the set

$$\begin{aligned}\{A_i, R_i, G_i, B_i, T_i, \overline{R}_i, \overline{G}_i\,\overline{B}_i, : \ 1 \le i \le n\} \ &\cup \{a_i : 1 \le i \le 2n+m+\lceil\log m\rceil + 11\} \cup \{c_i : 1 \le i \le 2n+1\} \\ &\cup \{d_i : 1 \le i \le \lceil\log m\rceil + 1\} \cup \{z_i : 2 \le i \le m + \lceil\log m\rceil + 6\} \\ &\cup \{A_{ij}, P_{ij}, \overline{P}_{ij}, R_{ij}, G_{ij}, B_{ij} : 1 \le i < j \le n\} \cup \{b, D, \overline{D}, e, T, S, N, \flat, \texttt{yes}, \texttt{no}\}.\end{aligned}$$

- $\Sigma(n) = \{A_{ij} : 1 \le i < j \le n\}$
- $w_1 = \{\{a_1, b, c_1, \texttt{yes}, \texttt{no}\}\}$
- $w_2(n) = \{\{D, A_1, \dots, A_n\}\}$
- $\mathcal{R}(\langle n, m\rangle)$ is the set of rules:
  (1) **Division rules:**
  $r_{1,i} \equiv [A_i]_2 \to [R_i]_2[T_i]_2$ for $i = 1, \dots, n$
  $r_{2,i} \equiv [T_i]_2 \to [G_i]_2[B_i]_2$ for $i = 1, \dots, n$
  (2) **Communication rules:**
  $r_{3,i} \equiv (1, a_i/a_{i+1}, 0)$ for $i = 1, \dots, 2n+m+\lceil\log m\rceil + 10$
  $r_{4,i} \equiv (1, c_i/c_{i+1}^2, 0)$ for $i = 1, \dots, 2n$
  $r_5 \equiv (1, c_{2n+1}/D, 2)$
  $r_6 \equiv (2, c_{2n+1}/d_1\overline{D}, 0)$
  $r_{7,i} \equiv (2, d_i/d_{i+1}^2, 0)$ for $i = 1, \dots, \lceil\log m\rceil$
  $r_8 \equiv (2, \overline{D}/e\,z_2, 0)$
  $r_{9,i} \equiv (2, z_i/z_{i+1}, 0)$ for $i = 2, \dots, m+\lceil\log m\rceil + 5$
  $r_{10,ij} \equiv (2, d_{\lceil\log m\rceil+1}A_{ij}/P_{ij}, 0)$ for $1 \le i < j \le n$
  $r_{11,ij} \equiv (2, P_{ij}/R_{ij}\overline{P}_{ij}, 0)$ for $1 \le i < j \le n$
  $r_{12,ij} \equiv (2, \overline{P}_{ij}/B_{ij}G_{ij}, 0)$ for $1 \le i < j \le n$
  $r_{13,ij} \equiv (2, R_iR_{ij}/R_i\overline{R}_j, 0)$ for $1 \le i < j \le n$
  $r_{14,ij} \equiv (2, B_iB_{ij}/B_i\overline{B}_j, 0)$ for $1 \le i < j \le n$
  $r_{15,ij} \equiv (2, G_iG_{ij}/G_i\overline{G}_j, 0)$ for $1 \le i < j \le n$
  $r_{16,j} \equiv (2, \overline{R}_jR_j/\flat, 0)$ for $1 \le j \le n$
  $r_{17,j} \equiv (2, \overline{B}_jB_j/\flat, 0)$ for $1 \le j \le n$
  $r_{18,j} \equiv (2, \overline{G}_jG_j/\flat, 0)$ for $1 \le j \le n$
  $r_{19} \equiv (2, e\,\flat/\lambda, 0)$
  $r_{20} \equiv (2, e\,z_{m+\lceil\log m\rceil+6}/T, 0)$
  $r_{21} \equiv (2, T/\lambda, 1)$
  $r_{22} \equiv (1, b\,T/S, 0)$
  $r_{23} \equiv (1, S\,\texttt{yes}/\lambda, 0)$
  $r_{24} \equiv (1, b\,a_{2n+m+\lceil\log m\rceil+11}/N, 0)$
  $r_{25} \equiv (1, N\,\texttt{no}/\lambda, 0)$
- $\mathcal{E}(\langle n, m\rangle) = \Gamma(\langle n, m\rangle) - \{\texttt{yes}, \texttt{no}\}$
- $i_{in} = 2$ is the *input cell.*
- $i_0 = 0$ is the *output region.*

5.1. An overview of the computation

First of all we define a polynomial encoding of the 3-coloring problem in the family $\mathbf{\Pi}$ constructed in the previous section. Let $u = (V, E)$ be an instance of the problem, with n vertices and m edges. Then we consider a *size* mapping on the set of instances defined as $s(u) = \langle n, m\rangle$. The encoding of the instance will be the multiset $cod(u) = \{\{A_{ij} : \{A_i, A_j\} \in E \wedge 1 \le i < j \le n\}\}$.

Next we informally describe how the recognizer tissue P system with cell division $\Pi(s(u))$ with input $cod(u)$ works.

Let us start with the *generation stage*. In this stage we have two parallel processes.

- On the one hand, in the cell labelled by 1 we have two counters: a_i, which will be used in the output stage, and c_i, which will be multiplied until step $2n$, where 4^n copies of c_{2n+1} are obtained.

- On the other hand, in the cell labelled by 2, the division rules are applied. For each object A_i (which encodes the i-th vertex of the graph) we get (in two steps) three cells labelled by 2, each of them encoding one of the three colors (*red, green* or *blue*) for this node by means of the objects R_i, G_i, B_i.

  After the appropriate divisions, in the step $2n$ we get exactly 3^n cells encoding all the possible 3-colorings of the graph.

In this way, after the $2n$-th step the generation stage is finished and the *checking stage* starts. At this moment, the content of the cell labelled by 1 is $\{\{a_{2n+1}, c_{2n+1}^{4^n}, b, \mathtt{yes}, \mathtt{no}\}\}$, and there are 3^n cells labelled by 2, each of them containing the object D, the objects A_{ij}, and encoding a different coloring function from V to $\{R, G, B\}$.

In the step $2n+1$, 3^n copies of c_{2n+1} in the cell 1 are traded against 3^n objects D, one from each cell labelled by 2. Notice that $4^n - 3^n$ spare copies of the counter c will remain in cell 1. When the object c_{2n+1} arrives to a cell labelled by 2, the communication process starts in that cell.

At the beginning of the process, we pay attention to the counters d and z. The first one will be multiplied until at least m copies are obtained, so that they can cooperate with the m input symbols A_{ij} that represent the edges. This is achieved in the $(2n + \lceil \log m \rceil + 2)$-th step (for the sake of simplicity, we shall denote $\gamma = 2n + \lceil \log m \rceil + 2$). The object z will be used to send an object T to the cell 1 at the end of this stage.

When the m copies of the object d are obtained, each of them is sent together with an object A_{ij} to the environment, and the corresponding objects P_{ij} are brought in. Then they are interchanged by R_{ij}, G_{ij}, B_{ij} from the environment (this is done in two steps by applying rules $r_{11,ij}$ and $r_{12,ij}$).

In order to know if a 3-coloring is valid we must check for each cell labelled by 2 (encoding a 3-coloring of the graph) if there exist two adjacent vertices with the same color. Let us reason with a color, say *red* (for *green* and *blue*, the process is the same):

- If node i has red color by a 3-coloring encoded in a cell labelled by 2, then the object R_i is present in that cell. Then in the step $\gamma + 3$ the objects R_i and R_{ij} produce $R_i\overline{R}_j$ (the object $\overline{R}_j$ is brought from the environment). Simultaneously, in that step the objects $\overline{P}_{ij}$ produce objects B_{ij} and G_{ij} applying the rules $r_{12,ij}$.
- If the vertex j is also of color *red*, then the objects $R_j\overline{R}_j$ are traded against an object $\flat$ from the environment. This object will be sent out in the next step, together with object e.
- In this way, as soon as the cell finds two adjacent vertices with the same color, it “rejects” its associated coloring by sending out the object e. Conversely, if there is a cell whose coloring is valid, then at the end of the checking stage the object e will still be present in that cell.

Notice that in the generation stage, the processes are carried out in parallel, but in the checking stage, the red color starts to be checked one step before the other two colors.

Taking into account that in the worst case there exists a node adjacent with m nodes (m is the number of edges), then after $\gamma + m + 3$ steps no more rules of the types $r_{13,ij}$, $r_{14,ij}$, and $r_{15,ij}$ can be applied for any $1 \leq i < j \leq n$. The checking stage will finish at the step $\gamma + m + 5$, when we can ensure that no more rules of the types $r_{16,j}$, $r_{17,j}$, and $r_{18,j}$ can be applied.

The output stage starts in the $(\gamma + m + 6)$-th step.

- *Affirmative answer*: If there exists a valid 3-coloring of the graph, then in some cell labelled by 2 from the configuration $\mathcal{C}_{\gamma+m+5}$ we have the object e and the object $z_{\lceil \log m \rceil + m + 6}$. By applying the rule r_{20} we produce an object T in that cell. In the next step, an object T arrives to the cell 1 by the application of the rule r_{21}. Then, the objects b and T in cell 1 permit that an object S arrives to cell 1 from the environment (by applying the rule r_{22}). Finally, an object $\mathtt{yes}$ is sent out to the environment by the application of the rule r_{23} in the step $\gamma + m + 9 = 2n + \lceil \log m \rceil + m + 11$. The obtained configuration is a halting one.
- *Negative answer*: If there is no valid 3-coloring of the graph, then the object e does not appear in any cell labelled by 2 from the configuration $\mathcal{C}_{\gamma+m+5}$, and actually no rule can be applied anymore in these cells. In the next three steps, only the rules for counter a in cell 1 are applied, and hence, at configuration $\mathcal{C}_{\gamma+m+8}$ we have the objects b and $a_{2n+\lceil \log m \rceil + m + 11}$ in cell 1. Next, by applying the rule r_{24} we get from the environment an object N, and in the following step an object $\mathtt{no}$ is sent out to the environment (recall that object $\mathtt{no}$ is present since the beginning of the computation in cell 1). The computation finishes in the step $\gamma + m + 10 = 2n + \lceil \log m \rceil + m + 12$.

5.2. Verification

Next, we prove that the family built above solves the 3-COL problem in a linear time, according to Definition 1.

First of all, the Definition requires that the defined family is *consistent*, in the sense that all systems of the family must be recognizer tissue P systems with cell division. By construction (type of rules and working alphabet) it is clear that it is a family of tissue P systems with cell division. In order to show that all members in $\mathbf{\Pi}$ are recognizer systems it suffices to check that all the computations halt (this will be deduced from the polynomial boundness), and that either an object $\mathtt{yes}$ or an object $\mathtt{no}$ is sent out exactly in the last step of the computation (this will be deduced from the soundness and completeness).

5.2.1. Polynomial uniformity of the family

Next, we show that the family $\mathbf{\Pi} = \{\Pi(\langle n, m\rangle) \ : \ n, m \in \mathbb{N}\}$ defined above is polynomially uniform by Turing machines. To this aim we are going to prove that it is possible to build $\Pi(\langle n, m\rangle)$ in polynomial time with respect to the size of u.

It is easy to check that the rules of a system $\Pi(\langle n, m\rangle)$ of the family are defined recursively from the values n and m. Besides, the necessary resources to build an element of the family are of a polynomial order, as shown below:

- Size of the alphabet: $3n^2 + 9n + 2m + 3\lceil\log m\rceil + 28 \in \theta(n^2 + m)$.
- Initial number of cells: $2 \in \theta(1)$.
- Initial number of objects: $n + m + 6 \in \theta(n + m)$.
- Number of rules: $18n^2 - 9n + 2m + 3\lceil\log m\rceil + 24 \in \theta(n^2 + m)$.
- Maximal length of a rule: 4.

Therefore, a deterministic Turing machine can build $\Pi(\langle n, m\rangle)$ in a polynomial time with respect to n and m. Besides, recall that $m \leq n^2$, since m is the number of edges in a graph of n vertices.

It is also interesting to bear in mind that every instance $u = (V, E)$ is introduced in the initial configuration of its associated cellular system via an input multiset (i.e. an 1-ary representation) and hence, $|u| \in O(n + m)$ holds.

We would like also to recall that the functions *cod* and *s* have been defined above for an instance $u = (V, E)$ of the problem 3-COL as follows: $cod(u) = \{\{A_{ij} : \{A_i, A_j\} \in E \wedge 1 \leq i < j \leq n\}\}$, and $s(u) = \langle n, m\rangle$, respectively. Both functions are computable in polynomial time and the pair (cod, s) is a polynomial encoding of $I_{3-\text{COL}}$ in Π, since for each instance u of the problem 3-COL we have that $cod(u)$ is an input multiset of the system $\Pi(s(u))$.

Next, following the indications of Definition 1, we shall prove that the family is polynomially bounded, and also that it is sound and complete with respect to (3-*COL*, *cod*, *s*).

5.2.2. Polynomial boundness of the family

In order to ensure that the system $\Pi(s(u))$ with input $cod(u)$ is polynomially (indeed, linearly) bounded, it suffices to find the moment in which the computation halts, or at least, an upper bound for it. As we shall show, the number of steps of the computations of any system of the family can always be bounded by a linear function. Nonetheless, we would like to stress that the amount of pre-computed resources for each instance u is polynomial in the size of the instance, since $cod(u)$ needs to be computed and $\Pi(s(u))$ needs to be built.

Proposition 2. *The family* $\mathbf{\Pi} = \{\Pi(\langle n, m\rangle) \ : \ n, m \in \mathbb{N}\}$ *is polynomially bounded with respect to* (3-COL, *cod*, *s*).

Proof (*Sketch*). We will informally go through the stages of the computation in order to estimate a bound for the number of steps. The computation will be studied more in detail when addressing the soundness and completeness proof.

Let $u = (V, E)$ be an instance of the problem 3-COL. We shall study what happens during the computation of the system $\Pi(s(u))$ with input $cod(u)$ which processes such instance in order to find the halting step, or at least, an upper bound for it.

First, the generation stage has exactly $2n$ steps, where all the divisions of the cells of the system are performed. The order in which the divisions are performed is nondeterministically chosen in each computation, but in all cases the divisions are carried out in the $2n$ first steps.

The pre-checking stage starts with the rule r_5. After three more steps, objects z_2 arrive to all cells 2, and the counter z starts working until it reaches its last (greatest) index, at the $(2n + \lceil\log m\rceil + m + 5)$-th step, and the checking stage ends.

The last one is the answer stage. The longest case is obtained when the answer is negative. In this case there is one step where only the counter a is working since no element T has reached the cell 1. In the next step an object N is brought from the environment and, at last, in the $(2n + \lceil\log m\rceil + m + 11)$-th step, the object `no` is sent to the environment.

Therefore, there exists a linear bound (with respect to n and m) on the number of steps of the computation. □

5.2.3. Soundness and completeness of the family

In order to prove the soundness and completeness of the family $\mathbf{\Pi}$ with respect to (3-*COL*, *cod*, *s*), we shall prove that given an instance u of the problem 3-COL, the system $\Pi(s(u))$ with input $cod(u)$ sends out an object `yes` if and only if the answer to the problem for the considered instance u is affirmative and the object `no` is sent out otherwise. In both cases the answer will be sent to the environment in the last step of the computation.

Proposition 3. *The family* $\mathbf{\Pi} = \{\Pi(\langle n, m\rangle) \ : \ n, m \in \mathbb{N}\}$ *is sound and complete with respect to* (3-COL, *cod*, *s*).

Proof. In order to complete the proof we shall proceed through a number of auxiliary results. □

Remark 4. For the sake of simplicity in the notation, we shall consider a function ψ defined as follows:

$$(f, i) = \begin{cases} R_i & \text{if } f(v_i) = R \\ G_i & \text{if } f(v_i) = G \\ B_i & \text{if } f(v_i) = B \end{cases}$$

for each 3-coloring $f : \{v_1, \ldots, v_n\} \to \{R, G, B\}$, and for $i = 1, \ldots, n$.

Given a computation $\mathcal{C}$ we denote the configuration at the i-th step as $\mathcal{C}_i$. Moreover, $\mathcal{C}_i(1)$ will denote the multiset associated to cell 1 in such configuration.

We start with the generation stage (i.e. the first $2n$ steps of the computation). It has two parallel processes, each of them in one cell.

Lemma 5. *Let $\mathcal{C}$ be an arbitrary computation of the system, then for all k such that* $0 \le k \le n$, $\mathcal{C}_{2k}(1) = \{\{a_{2k+1}, c_{2k+1}^{4^k}, b, \texttt{yes}, \texttt{no}\}\}$ *holds.*

Proof. We shall reason by induction on k.
Base Case. We have $\mathcal{C}_0(1) = \{\{a, c_1, b, \texttt{yes}, \texttt{no}\}\}$, and thus the lemma holds for $k = 0$.
Case $k < n \to k+1$. Let k be such that $1 \le k < n$ and we have, by inductive hypothesis, $\mathcal{C}_{2k}(1) = \{\{a_{2k+1}, c_{2k+1}^{4^k}, b, \texttt{yes}, \texttt{no}\}\}$. In this configuration, only the rules $r_{3,k+1}$ and $r_{4,k+1}$ can be applied to cell 1, and therefore $\mathcal{C}_{2k+1}(1) = \{\{a_{2k+2}, c_{2k+2}^{2\cdot 4^k}, b, \texttt{yes}, \texttt{no}\}\}$. In the following step, we can only apply to cell 1 the rules $r_{3,k+2}$ and $r_{4,k+2}$, and we thus obtain

$$\mathcal{C}_{2k+2}(1) = \{\{a_{2k+3}, c_{2k+3}^{2\cdot 2\cdot 4^k}, b, \texttt{yes}, \texttt{no}\}\} = \{\{a_{2k+3}, c_{2k+3}^{4^{k+1}}, b, \texttt{yes}, \texttt{no}\}\}. \quad \square$$

Proposition 6. *Let $\mathcal{C}$ be an arbitrary computation of the system, then:*

(1) *For each coloring $f : \{v_1, \dots, v_n\} \to \{R, B, G\}$ there exists only one cell 2 in $\mathcal{C}_{2n}$ whose multiset is $\{\{A_{ij} : \{v_i, v_j\} \in E\}\} \cup \{\{D, \psi(f, 1), \dots, \psi(f, n)\}\}$*
(2) *There exist exactly 3^n cells labelled by 2 in configuration $\mathcal{C}_{2n}$.*

Proof (*Sketch*). Let f be a coloring $f : \{v_1, \dots, v_n\} \to \{R, B, G\}$, and let us denote by R_{f_1} and R_{f_2} the following sets of rules

$$R_{f_1} = \{r_{1,i} : f(v_i) = R, \quad 1 \le i \le n\}$$
$$R_{f_2} = \{r_{1,i}, r_{2,i} : f(v_i) = B \vee f(v_i) = G, \quad 1 \le i \le n\}.$$

$R_f = R_{f_1} \cup R_{f_2}$ is the set of rules needed to produce a cell 2 containing the objects $\psi(f, 1), \dots, \psi(f, n)$ that encode the coloring f. More precisely, for each $1 \le i \le n$, if $f(v_i) = R$ then after applying $r_{1,i} \in R_{f_1}$ we take the cell where R_i occurs, and if $f(v_i) \ne R$, then after applying $r_{1,i} \in R_{f_2}$ we take the cell where T_i occurs, and after applying $r_{2,i} \in R_{f_2}$ we take the cell where $\psi(f, i)$ occurs. One can see that $|R_f| = q + 2(n - q)$ where $q = |R_{f_1}|$, and it is clear that rules in R_f cannot be applied in parallel over the same cell, as all of them are division rules. Thus, it takes $2n - q$ steps to carry them out sequentially (note that $1 \le q \le n$).

Due to the intrinsic non-determinism of the system, a detailed proof falls out of the scope of this paper. Informally, the idea is that rules from R_f may be applied in different orders within different computations, but it can be shown that in every computation after $2n$ steps no more division rules are applied in any cell 2. The design of the rules ensures that each one of the 3^n possible colorings will be represented by some cell in the system. $\square$

Lemma 7. *Let $\mathcal{C}$ be an arbitrary computation of the system, then for all i such that* $1 \le i \le \lceil \log m \rceil + m + 8$, $\mathcal{C}_{2n+i}(1) = \{\{a_{2n+i+1}, c_{2n+1}^{4^n - 3^n}, D^{3^n}, b, \texttt{yes}, \texttt{no}\}\}$ *holds.*

Proof. In order to prove the lemma it suffices to observe the following:

- $\mathcal{C}_{2n}(1) = \{\{a_{2n+1}, c_{2n+1}^{4^n}, b, \texttt{yes}, \texttt{no}\}\}$ holds from Lemma 5.
- There exist exactly 3^n cells labelled by 2 in configuration $\mathcal{C}_{2n}$, each of them containing an object D (follows from previous result).
- In the next step of the computation, only rules r_5 and $r_{3,2n+1}$ are applicable on cell 1, yielding $\mathcal{C}_{2n+1}(1) = \{\{a_{2n+1+1}, c_{2n+1}^{4^n - 3^n}, D^{3^n}, b, \texttt{yes}, \texttt{no}\}\}$
- During the rest of the checking stage, only rules of type $r_{3,i}$ are applicable on cell 1, and the result follows. $\square$

Proposition 8. *Let $\mathcal{C}$ be an arbitrary computation of the system. Then:*

- *For each coloring $f : \{v_1, \dots, v_n\} \to \{R, B, G\}$ there exists only one cell 2 in $\mathcal{C}_{2n+1}$ whose associated multiset is*

$$\{\{A_{ij} : \{v_i, v_j\} \in E\}\} \cup \{\{c_{2n+1}, \psi(f, 1), \dots, \psi(f, n)\}\}.$$

- *There exist exactly 3^n cells labelled by 2 in configuration $\mathcal{C}_{2n+1}$*

Proof. $\mathcal{C}_{2n+1}$ is obtained from $\mathcal{C}_{2n}$ by the application of the rules r_5 and $r_{3,n+1}$ and hence, 3^n objects c_{2n+1} in the cell 1 are traded against 3^n objects D from the cells 2 (one from each cell). Then $\mathcal{C}_{2n+1}(1) = \{\{a_{2n+2}, c_{2n+1}^{4^n - 3^n}, D^{3^n}, b, \texttt{yes}, \texttt{no}\}\}$ and for every $f : \{v_1, \dots, v_n\} \to \{R, B, G\}$ there exists only one cell 2 whose associated multiset is $\{\{A_{ij} : \{v_i, v_j\} \in E\}\} \cup \{\{c_{2n+1}, \psi(f, 1), \dots, \psi(f, n)\}\}$.

Since no division rule has been applied in this step (actually, they will not be applied anymore along the computation), the number of cells 2 remains the same as in the previous configuration. $\square$

Proposition 9. *Let $\mathcal{C}$ be an arbitrary computation of the system. Then:*

- *For each $f : \{v_1, \dots, v_n\} \to \{R, B, G\}$ there exists only one cell 2 in $\mathcal{C}_{2n+2}$ whose associated multiset is*

$$\{\{A_{ij} : \{v_i, v_j\} \in E\}\} \cup \{\{d_1, \overline{D}, \psi(f, 1), \dots, \psi(f, n)\}\}.$$

- *There exist exactly 3^n cells labelled by 2 in configuration $\mathcal{C}_{2n+2}$.*

Proof. It follows from the previous result, taking into account that $\mathcal{C}_{2n+2}$ is obtained from $\mathcal{C}_{2n+1}$ by applying rules r_6 and $r_{3,2n+2}$. Therefore, in cell 1 only the object a_{2n+2} is replaced by a_{2n+3}, and each cell 2 trades an object c_{2n+1} against the objects $d_1, \overline{D}$ from the environment. $\square$

Proposition 10. *Let $\mathcal{C}$ be an arbitrary computation of the system. For each i $(1 \le i \le \lceil \log m \rceil)$ and for each $f : \{v_1, \ldots, v_n\} \to \{R, B, G\}$ there exists only one cell 2 in $\mathcal{C}_{2n+2+i}$ whose associated multiset is*

$$\{\{A_{ij} : \{v_i, v_j\} \in E\}\} \cup \{\{d_{i+1}^{2^i}, e, z_{i+1}, \psi(f, 1), \ldots, \psi(f, n)\}\}.$$

Proof. We shall reason by induction on i.
Case $i = 1$. The result follows from the previous Proposition, as $\mathcal{C}_{2n+2+1}$ is obtained from $\mathcal{C}_{2n+2}$ by applying the rules $r_{3,2n+3}$, $r_{7,1}$ and r_8.
Case $1 \le i < \lceil \log m \rceil \to i + 1$. Let i be such that $1 \le i < \lceil \log m \rceil$ and let us suppose that the result holds for i. The configuration $\mathcal{C}_{2n+2+i+1}$ is obtained from the configuration $\mathcal{C}_{2n+2+i}$ via the application of the rules $r_{3,2n+2+i}$, $r_{7,i+1}$ and $r_{9,i}$. By applying the rule $r_{3,2n+2+i}$ to $\mathcal{C}_{2n+2+i}$, the object a_{2n+3+i} (used as a counter in the cell 1) evolves by adding 1 to the subscript. The application of the rule $r_{7,i+1}$ in all cells 2 replaces each object d_{i+1} by d_{i+2}^2. Finally, the rule $r_{9,i}$ is applied in each cell 2 over the object z_{i+1} (used as a counter). Therefore,

$$\mathcal{C}_{2n+3+i}(1) = \{\{a_{2n+3+i}, c_{2n+1}^{4^n-3^n}, D^{3^n}, b, \texttt{yes}, \texttt{no}\}\}.$$

Bearing in mind that, by inductive hypothesis, for each $f : \{v_1, \ldots, v_n\} \to \{R, B, G\}$ there exists only one cell 2 in $\mathcal{C}_{2n+2+i}$ with the multiset

$$\{\{A_{ij} : \{v_i, v_j\} \in E\}\} \cup \{\{d_{i+1}^{2^i}, e, z_{i+1}, \psi(f, 1), \ldots, \psi(f, n)\}\},$$

then the multiset of such cell 2 in $\mathcal{C}_{2n+3+i}$ will be

$$\{\{A_{ij} : \{v_i, v_j\} \in E\}\} \cup \{\{d_{i+2}^{2^{i+1}}, e, z_{i+2}, \psi(f, 1), \ldots, \psi(f, n)\}\}. \quad \square$$

Proposition 11. *Let $\mathcal{C}$ be an arbitrary computation of the system. For each $f : \{v_1, \ldots, v_n\} \to \{R, B, G\}$ there exist some unique cells 2 in the configurations $\mathcal{C}_{2n+\lceil \log m \rceil+3}$ and $\mathcal{C}_{2n+\lceil \log m \rceil+4}$ with the following corresponding multisets:*

$$\{\{P_{ij} : \{v_i, v_j\} \in E\}\} \cup \{\{e, z_{\lceil \log m \rceil+2}, \psi(f, 1), \ldots, \psi(f, n)\}\}$$
$$\{\{R_{ij}, \overline{P}_{ij} : \{v_i, v_j\} \in E\}\} \cup \{\{e, z_{\lceil \log m \rceil+3}, \psi(f, 1), \ldots, \psi(f, n)\}\}.$$

Proof. The configuration $\mathcal{C}_{2n+\lceil \log m \rceil+3}$ is obtained from $\mathcal{C}_{2n+\lceil \log m \rceil+2}$ by applying the rules $r_{3,2n+\lceil \log m \rceil+3}$, $r_{9,\lceil \log m \rceil+1}$, $\{r_{10,ij} : \{v_i, v_j\} \in E\}$.

- $r_{3,2n+\lceil \log m \rceil+3}$ allows the evolution of the counter $a_{2n+\lceil \log m \rceil+3}$ in the cell 1.
- $r_{9,\lceil \log m \rceil+1}$ allows the evolution of the counter $z_{\lceil \log m \rceil+2}$ in each cell 2.
- $\{r_{10,ij} : \{v_i, v_j\} \in E\}$ allow the replacement of the objects A_{ij} (encoding the edges) by P_{ij}.

By an analogous reasoning, we can check that $\mathcal{C}_{2n+\lceil \log m \rceil+4}$ is obtained from the configuration $\mathcal{C}_{2n+\lceil \log m \rceil+3}$ by application of the rules $r_{3,2n+\lceil \log m \rceil+4}$, $r_{9,\lceil \log m \rceil+2}$ and $\{r_{11,ij} : \{v_i, v_j\} \in E\}$. $\square$

Proposition 12. *Let $\mathcal{C}$ be an arbitrary computation of the system. Then for each $f : \{v_1, \ldots, v_n\} \to \{R, B, G\}$ there exist only one cell 2 in $\mathcal{C}_{2n+\lceil \log m \rceil+5}$ whose associated multiset is*

$$\{\{R_i, \overline{R}_j, G_{ij}, B_{ij} : \{v_i, v_j\} \in E \wedge f(v_i) = R\}\} \cup \{\{R_{ij}, G_{ij}, B_{ij} : \{v_i, v_j\} \in E \wedge f(v_i) \neq R\}\}$$
$$\cup \{\{\psi(f, j) : 1 \le j \le n \wedge f(v_j) \neq R\}\} \cup \{\{e, z_{\lceil \log m \rceil+4}\}\}.$$

Proof. The configuration $\mathcal{C}_{2n+\lceil \log m \rceil+5}$ is obtained from $\mathcal{C}_{2n+\lceil \log m \rceil+4}$ by applying the rules listed below:

- $r_{3,2n+\lceil \log m \rceil+5}$ allows the evolution of the counter $a_{2n+\lceil \log m \rceil+5}$ in the cell 1.
- $r_{9,\lceil \log m \rceil+3}$ allows the evolution of the counter $z_{\lceil \log m \rceil+3}$ in each cell 2.
- $\{r_{12,ij} : \{v_i, v_j\} \in E\}$ allow the replacement of the objects $\{\overline{P}_{ij} : \{v_i, v_j\} \in E\}$ by $\{B_{ij}, G_{ij} : \{v_i, v_j\} \in E\}$.
- $\{r_{13,ij} : \{v_i, v_j\} \in E \wedge f(i) = R\}$ produce, in each cell 2, pairs of objects $R_i, \overline{R}_j$ for each $\{v_i, v_j\} \in E$ such that $f(i) = R$. Objects R_{ij} corresponding to edges $\{v_i, v_j\} \in E$ such that $f(i) \neq R$ will remain unchanged. $\square$

Proposition 13. *Let $\mathcal{C}$ be an arbitrary computation of the system. Then*

- *There exist exactly 3^n cells labelled by 2 in configuration $\mathcal{C}_{2n+\lceil \log m \rceil+m+6}$.*
- *If $f : \{v_1, \ldots, v_n\} \to \{R, B, G\}$ is not a valid coloring, then there exists only one cell 2 in $\mathcal{C}_{2n+m+\lceil \log m \rceil+6}$ associated with f, and whose associated multiset is*

  $$\{\{z_{m+\lceil \log m \rceil+5}\}\} \cup \{\{\flat^{t-1}\}\}$$

  together with an irrelevant multiset of objects, where t is the number of adjacent pairs of nodes with the same color in f.

- *If $f : \{v_1, \ldots, v_n\} \to \{R, B, G\}$ is a valid coloring, then there exists only one cell 2 in $\mathcal{C}_{2n+m+\lceil \log m\rceil+6}$ associated with f, and whose associated multiset is*

  $$\{\{e, z_{m+\lceil \log m\rceil+5}\}\}$$

  together with an irrelevant multiset of objects (not containing $\flat$).

Proof. The check for the validity of the coloring started in the configuration $\mathcal{C}_{2n+\lceil \log m\rceil+5}$ for the color R.

From $\mathcal{C}_{2n+\lceil \log m\rceil+6}$ on, such a check is simultaneous in each cell 2 with the check of the colors B and G. More precisely, in $\mathcal{C}_{2n+\lceil \log m\rceil+5+i}(1)$, with $1 \le i \le m+1$, only the counter $a_{2n+\lceil \log m\rceil+6+i}$ evolves via the application of the rule $r_{3,2n+\lceil \log m\rceil+6+i}$.

The check for the validity of the colorings is performed in each cell 2, and it is carried out by the rules $\{r_{l,ij} : 13 \le l \le 18 \wedge \{v_i, v_j\} \in E \wedge i < j\}$.

Let us consider a coloring f. When the objects R_{ij}, B_{ij}, G_{ij} have been generated for each $\{v_i, v_j\} \in E$, $(i < j)$, then at most m steps will be necessary for all checks to be concluded. Indeed, given a node v_i (assume e.g. that $f(v_i) = R$), each edge $\{v_i, v_j\} \in E$, $(i < j)$, will be checked only once by the rule $(2, R_iR_{ij}/R_i\overline{R}_j, 0)$, and analogously if $f(v_i) \in \{G, B\}$.

In order to give enough time for all checks to be accomplished (recall that the edges are checked in a nondeterministic order), the counter z evolves in that stage from $z_{\lceil \log m\rceil+1}$ till $z_{m+\lceil \log m\rceil+5}$. Notice that we need $m+1$ steps, because objects G_{ij} and B_{ij} appear one step later than R_{ij}.

Besides, by the application of the rules $r_{l,ij}$ and $r_{l+3,ij}$, $(13 \le l \le 15)$ in each cell, an object $\flat$ is generated for each edge connecting two nodes with the same color in the corresponding coloring. Thus, this cell in $\mathcal{C}_{2n+m+\lceil \log m\rceil+6}$ will not contain object e because it has been sent out together with an object $\flat$ by means of the application of the rule r_{19}. If the object $\flat$ does not appear in the corresponding cell for a given coloring, this will mean that this is a valid coloring. □

Proposition 14. *Let $\mathcal{C}$ be an arbitrary computation of the system.*

(1) *For each coloring f which is valid, there exists one cell 2 in $\mathcal{C}_{2n+m+\lceil \log m\rceil+7}$ that encodes f and contains the objects e, $z_{m+\lceil \log m\rceil+6}$. Besides this cell in $\mathcal{C}_{2n+m+\lceil \log m\rceil+8}$ contains the object T and does not contain the object e.*
(2) *For each coloring f which is not valid, there exists one cell 2 in the configuration $\mathcal{C}_{2n+m+\lceil \log m\rceil+7}$ that encodes f and contains the object $z_{m+\lceil \log m\rceil+6}$, but does not contain the object e. Besides, this cell in $\mathcal{C}_{2n+m+\lceil \log m\rceil+8}$ still contains the object $z_{m+\lceil \log m\rceil+6}$ but does not contain the object T.*

Proof.
- If f is valid, then the object $\flat$ does not appear in the cell associated with f in $\mathcal{C}_{2n+m+\lceil \log m\rceil+6}$, and thus only the object $z_{m+\lceil \log m\rceil+5}$ evolves in such cell 2. Therefore, the statement (1) follows directly from the previous proposition.
- If f is not valid, then there exists in $\mathcal{C}_{2n+m+\lceil \log m\rceil+6}$ a cell 2 associated with it containing $t-1$ objects $\flat$. On the other hand, the object $z_{m+\lceil \log m\rceil+5}$ is present in every cell 2 from $\mathcal{C}_{2n+m+\lceil \log m\rceil+6}$, and hence the first statement from (2) follows. Finally, in $\mathcal{C}_{2n+m+\lceil \log m\rceil+8}$ this cell will not contain object T, because rule r_{20} could not be applied in the absence of object e. □

Proposition 15. *Let $\mathcal{C}$ be an arbitrary computation of the system. Let us suppose that there exists a valid coloring of the graph with 3 colors. Then*

(a) $\mathcal{C}_{2n+m+\lceil \log m\rceil+9}(1) = \{\{a_{2n+m+\lceil \log m\rceil+10}, c_{2n+1}^{4^n-3^n}, D^{3^n}, b, \mathtt{yes}, \mathtt{no}, T^\alpha\}\}$
(b) $\mathcal{C}_{2n+m+\lceil \log m\rceil+10}(1) = \{\{a_{2n+m+\lceil \log m\rceil+11}, c_{2n+1}^{4^n-3^n}, D^{3^n}, b, \mathtt{yes}, \mathtt{no}, S, T^{\alpha-1}\}\}$
(c) $\mathcal{C}_{2n+m+\lceil \log m\rceil+11}(1) = \{\{a_{2n+m+\lceil \log m\rceil+11}, c_{2n+1}^{4^n-3^n}, D^{3^n}, \mathtt{no}, T^{\alpha-1}\}\}$

where α is the number of existing valid colorings. Furthermore, the object $\mathtt{yes}$ *appears in $\mathcal{C}_{2n+m+\lceil \log m\rceil+11}(0)$.*

Proof. The configuration of item (a) is obtained by the application of rules r_{21} and $r_{3,2n+m+\lceil \log m\rceil+9}$ to the previous configuration. Analogously, the configurations of items (b) and (c) are obtained by the application of rules r_{22} and r_{23} respectively. □

Proposition 16. *Let $\mathcal{C}$ be an arbitrary computation of the system. Let us suppose that there does not exist a valid coloring of the graph with 3 colors. Then*

(a) $\mathcal{C}_{2n+m+\lceil \log m\rceil+9}(1) = \{\{a_{2n+m+\lceil \log m\rceil+10}, c_{2n+1}^{4^n-3^n}, D^{3^n}, b, \mathtt{yes}, \mathtt{no}\}\}$
(b) $\mathcal{C}_{2n+m+\lceil \log m\rceil+10}(1) = \{\{a_{2n+m+\lceil \log m\rceil+11}, c_{2n+1}^{4^n-3^n}, D^{3^n}, b, \mathtt{yes}, \mathtt{no}\}\}$
(c) $\mathcal{C}_{2n+m+\lceil \log m\rceil+11}(1) = \{\{N, c_{2n+1}^{4^n-3^n}, D^{3^n}, \mathtt{yes}, \mathtt{no}\}\}$
(d) $\mathcal{C}_{2n+m+\lceil \log m\rceil+12}(1) = \{\{c_{2n+1}^{4^n-3^n}, D^{3^n}, \mathtt{yes}\}\}$.

Furthermore, the object $\mathtt{no}$ *appears in $\mathcal{C}_{2n+m+\lceil \log m\rceil+12}(0)$.*

Proof. The configuration of item (a) is obtained by the application of rule $r_{3,2n+m+\lceil \log m\rceil+9}$ to the previous configuration. Then, by applying the rule $r_{3,2n+m+\lceil \log m\rceil+10}$ we obtain

$$\mathcal{C}_{2n+m+\lceil \log m\rceil+10}(1) = \{\{a_{2n+m+\lceil \log m\rceil+11}, c_{2n+1}^{4^n-3^n}, D^{3^n}, b, \mathtt{yes}, \mathtt{no}\}\}$$

by applying the rule r_{24} we obtain

$$\mathcal{C}_{2n+m+\lceil\log m\rceil+11}(1) = \{\{N, c_{2n+1}^{4^n-3^n}, D^{3^n}, \mathtt{yes}, \mathtt{no}\}\}$$

by applying the rule r_{25} we obtain

$$\mathcal{C}_{2n+m+\lceil\log m\rceil+12}(1) = \{\{c_{2n+1}^{4^n-3^n}, D^{3^n}, \mathtt{yes}\}\}$$

and $\mathtt{no} \in \mathcal{C}_{2n+m+\lceil\log m\rceil+12}(0)$. □

5.3. Main results

From the discussion in the previous sections and according to the definition of solvability given in Section 4, we deduce the following result:

Theorem 17. 3-*coloring* $\in$ $\mathbf{PMC}_{TD}$.

As a consequence of this result we have:

Theorem 18. $\mathbf{NP} \cup \mathbf{co} - \mathbf{NP} \subseteq \mathbf{PMC}_{TD}$.

Proof. It suffices to make the following observations: the 3-coloring problem is **NP**-complete, 3-coloring $\in \mathbf{PMC}_{TD}$ and the class $\mathbf{PMC}_{TD}$ is stable under polynomial-time reduction, and also closed under complement. □

6. Final remarks and future work

The power and efficiency of cell-like P systems for solving **NP**-complete problems have been widely studied (in the framework of cell division and membrane creation). Nevertheless, there are very few works studying the case of tissue-like P systems.

We would like to stress that until now there there has been no methodology for achieving the formal verification of tissue P systems based cellular solutions to problems. Indeed, it seems that the formal verification process is hard to achieve due to the massive parallelism and non-determinism of the model.

In this paper we have proposed a new solution to an **NP**-complete problem, the 3-coloring problem, which can be used as a scheme for designing solutions to other **NP**–complete problems from graph theory such as the *vertex–cover* problem, the *clique* problem, the *Hamiltonian path* problem, etc. Moreover, the structure of the solution described can be also adapted for solving computationally hard numerical problems.

Recently, a new kind of P system model (called spiking neural P systems) based on the idea of spiking neurons has been presented (see, for example, [10]). The motivation for this model comes from two directions: the attempt of membrane computing to pass from cell-like architectures to tissue-like or neural-like architectures (see [22], [18]), and the intriguing possibility of encoding information in the duration of events, or in the interval of time elapsed between events, as in recent research in neural computing (of *third generation*) [14].

Until now, spiking neural P systems have been basically used in the generative mode and the investigations have been addressed to study the computational completeness of these models. It remains as further work to bridge tissue P systems and this new model in order to be able to solve **NP**–complete problems through spiking neural P systems.

Acknowledgement

The authors wish to acknowledge the support of the project TIN2005-09345-C04-01 of the Ministerio de Educación y Ciencia of Spain, cofinanced by FEDER funds, and the support of the project of excellence TIC-581 of the Junta de Andaluca.

References

[1] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, P. Walter, The Molecular Biology of the Cell, fourth ed., Garland Publ. Inc., London, 2002.
[2] A. Alhazov, R. Freund, M. Oswald, Tissue P Systems with Antiport Rules ans Small Numbers of Symbols and Cells, in: Lecture Notes in Computer Science, vol. 3572, 2005, pp. 100–111.
[3] K. Appel, W. Haken, Every planar map is 4-colorable - 1: Discharging, Illinois Journal of Mathematics 21 (1977) 429–490.
[4] K. Appel, W. Haken, Every planar map is 4-colorable - 2: Reducibility, Illinois Journal of Mathematics 21 (1977) 491–567.
[5] F. Bernardini, M. Gheorghe, Cell communication in tissue P systems and cell division in population P systems, Soft Computing 9 (9) (2005) 640–649.
[6] L. Cardelli, Brane Calculi, in: Lecture Notes in Computer Science, vol. 3082, 2005, pp. 257–278.
[7] R. Freund, Gh. Păun, M.J. Pérez-Jiménez, Tissue P systems with channel states, Theoretical Computer Science 330 (2005) 101–116.
[8] M.R. Garey, D.S. Johnson, Computers and Intractability A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, 1979.
[9] M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, F.J. Romero-Campero, A Linear Solution for QSAT with Membrane Creation, in: Lecture Notes in Computer Science, vol. 3850, 2006, pp. 241–252.
[10] M. Ionescu, Gh. Păun, T. Yokomori, Spiking neural P systems, Fundamenta Informaticae 71 (2–3) (2006) 279–308.
[11] S.N. Krishna, K. Lakshmanan, R. Rama, Tissue P Systems with Contextual and Rewriting Rules, in: Lecture Notes in Computer Science, vol. 2597, 2003, pp. 339–351.
[12] K. Lakshmanan, R. Rama, On the power of tissue P systems with insertion and deletion rules, in: A. Alhazov, C. Martín-Vide and Gh. Păun (Eds.), Preproceedings of the Workshop on Membrane Computing, Tarragona, Report RGML 28/03, 2003, pp. 304–318.

[13] P.L. Luisi, The Chemical Implementation of Autopoiesis, in: G.R. Fleishaker, et al. (Eds.), Self-Production of Supramolecular Structures, Kluwer, Dordrecht, 1994.
[14] W. Maass, C. Bishop (Eds.), Pulsed Neural Networks, MIT Press, Cambridge, 1999.
[15] C. Martín Vide, J. Pazos, Gh. Păun, A. Rodríguez Patón, A New Class of Symbolic Abstract Neural Nets: Tissue P Systems, in: Lecture Notes in Computer Science, vol. 2387, 2002, pp. 290–299.
[16] C. Martín Vide, J. Pazos, Gh. Păun, A. Rodríguez Patón, Tissue P systems, Theoretical Computer Science 296 (2003) 295–326.
[17] Gh. Păun, Computing with membranes, Journal of Computer and System Sciences 61 (1) (2000) 108–143.
[18] Gh. Păun, Membrane Computing. An Introduction, Springer–Verlag, Berlin, 2002.
[19] A. Păun, Gh. Păun, The power of communication: P systems with symport/antiport, New Generation Computing 20 (3) (2002) 295–305.
[20] Gh. Păun, M.J. Pérez-Jiménez, Recent computing models inspired from biology: DNA and membrane computing, Theoria 18 (46) (2003) 72–84.
[21] Gh. Păun, M.J. Pérez-Jiménez, A. Riscos-Núñez, Tissue P system with cell division, in: Gh. Păun, A. Riscos-Núñez, A. Romero-Jiménez and F. Sancho-Caparrini (Eds.), Second Brainstorming Week on Membrane Computing, Sevilla, Report RGNC 01/2004, 2004, pp. 380–386.
[22] Gh. Păun, Y. Sakakibara, T. Yokomori, P systems on graphs of restricted forms, Publicationes Mathematicae Debrecen 60 (2002) 635–660.
[23] M.J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini, Teoría de la Complejidad en Modelos de Computación Celular con Membranas. Editorial Kronos, Sevilla, 2002.
[24] M.J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini, The P versus NP problem through cellular computing with membranes, in: Lecture Notes in Computer Science, vol. 2950, 2004, pp. 338–352.
[25] M.J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini, A polynomial complexity class in P systems using membrane division, in: E. Csuhaj-Varjú, C. Kintala, D. Wotschke and Gy. Vaszyl (Eds.), Proceedings of the 5th Workshop on Descriptional Complexity of Formal Systems, DCFS 2003, 2003, pp. 284–294.
[26] V.J. Prakash, On the power of tissue P systems working in the maximal-one mode, in: A. Alhazov, C. Martín-Vide and Gh. Păun (Eds.), Preproceedings of the Workshop on Membrane Computing, Tarragona, Report RGML 28/03, 2003, pp. 356–364.
[27] L.J. Stockmeyer, Planar 3-colorability is NP-complete, SIGACT News 5 (3) (1973) 19–25.
[28] C. Zandron, C. Ferretti, G. Mauri, Solving NP-complete problems using P systems with active membranes, in: I. Antoniou, C.S. Calude, M.J. Dinneen (Eds.), Unconventional Models of Computation, UMC'2K, Springer-Verlag, 2000, pp. 289–301.
[29] ISI web page. http://esi-topics.com/erf/october2003.html.
[30] P systems web page. http://psystems.disco.unimib.it/.