A Linear–time Tissue P System Based Solution for the 3–coloring Problem

Daniel Díaz-Pernil[1] Miguel A. Gutiérrez-Naranjo[2]
Mario J. Pérez-Jiménez[3] Agustín Riscos-Núñez[4]

Research Group on Natural Computing
Dpto. de Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, Spain

Abstract

In the literature, several examples of the efficiency of cell-like P systems regarding the solution of **NP**-complete problems in polynomial time can be found (obviously, trading space for time). Recently, different new models of tissue-like P systems have received important 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

Membranes 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].

This paper is enclosed in the Natural Computing framework. More precisely, in the study of the structure and functioning of cells as living organisms able to process and generate information. Assuming this starting point, two different disciplines within Natural Computing can be found in the literature: *Membrane Computing* and *Brane Calculi*.

[1] Email: sbdani@us.es
[2] Email: magutier@us.es
[3] Email: marper@us.es
[4] Email: ariscosn@us.es

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, 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 [19].

This emergent cross-disciplinary branch of Natural Computing was introduced by Gh. Păun in [18]. It has received important attention from the scientific community since then, with contributions by computer scientists, biologists, formal linguists and complexity theoreticians, enriching each others with results, open problems and promising new research lines. In fact, Membrane Computing has been selected by the Institute for Scientific Information, USA, as a fast *Emerging Research Front* in Computer Science, and [20] was mentioned in [30] 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 [5].

In the last 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.

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 [10] 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 in living cells, cell-like P systems abstract the way of obtaining new membranes, mainly from two biological processes: *mitosis* (membrane division) and *autopoiesis*, see [14] (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 pretty different. In fact, nowadays there does not exist any theoretical result which proves that these models can simulate each other in polynomial time.

Under the hypothesis **P**≠**NP**, Zandron et al. [29] 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. [25]

[5] A layman-oriented introduction can be found in [21] and further bibliography at [31].

obtaining a characterization of the **P≠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 (because of their membrane structure) *Tissue P Systems*. Instead of considering that membranes are hierarchically arranged, the membranes are placed in the nodes of a graph. This variant has two biological inspirations (see [17]): 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[6]. 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 [16,17], several research lines have been developed and other variants have arisen (see, for example, [2,5,7,12,13,27]). One of the most interesting variants of Tissue P systems was presented in [22]. 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 [22], a polynomial-time solution to the **NP**-complete problem SAT is shown. In this paper we go on with the research in this variant 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, with a short overview of the computation and the necessary resources. Finally, the main results, some conclusions and new open research lines are presented.

2 Preliminares

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 $f : A \to \mathbb{N}$ is a mapping. If

[6] This way of communication for P systems was introduced in [20].

$m = (A, f)$ is a multiset then its *support*, $supp(m)$, is defined as $supp(m) = \{x \in A \mid f(x) > 0\}$ 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, then it will be denoted as $m = \{\{a_1, \ldots, a_k\}\}$, where each element a_i occurs $f(a_i)$ times.

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.

A *path* of length $n \geq 1$ from a vertex x to a vertex y in a graph $G = (V, E)$ is a sequence $\{v_0, v_1, \ldots, v_n\}$ of vertices such that $v_0 = x$, $v_n = y$ and $\{\{v_i, v_{i+1}\} \mid i = 0, \ldots, n-1\} \subseteq E$. If there is a path from u to v in G, we will say that v is *reachable* from u in G and it will be denoted by $u \rightsquigarrow_G v$. Two vertices u and v are *connected* in G if $u \rightsquigarrow_G v$. A graph is *connected* if for every pair of different vertices from V, one of them is reachable from the other one.

In what follows we assume the reader is already familiar with the basic notions and the terminology underlying P systems. For details, see [19].

3 Tissue P Systems with Cell Division

In the first definition of the model of tissue P systems [16,17] the membrane structure did not change along the computation. Based on the cell-like model of P systems with active membranes, Gh. Păun et al. presented in [22] a new model 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 [19]); 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 the communication channels with other cells and with the environment.

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:

(i) Γ is a finite *alphabet*, whose symbols will be called *objects*.

(ii) $w_1, \ldots, w_q$ are strings over Γ.

(iii) $\mathcal{E} \subseteq \Gamma$.

(iv) $\mathcal{R}$ 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*: $[a]_i \rightarrow [b]_i[c]_i$, where $i \in \{1, 2, \ldots, q\}$ and $a, b, c \in \Gamma$.

(v) $i_0 \in \{0, 1, 2, \ldots, q\}$.

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 will use 0 to refer to the label of the environment, and i_0 denotes the output region (which can be the region inside a membrane or the environment).

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 $[a]_i \rightarrow [b]_i[c]_i$ can be applied over a cell i containing object a. The application of this rule divides this cell into other two 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 variant of tissue P systems with cell division is introduced in [22]: *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 [26] 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 if a string belongs or not 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 codifying 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.

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

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

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 `yes` and `no`, present in at least one copy in w_1, w_2, ..., 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_o is the environment.
- All computations halt.
- If $\mathcal{C}$ is a computation of Π, then either the object `yes` or the object `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 $w \in \Gamma^*$ 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 `yes` is sent to the environment, in the last step of the corresponding computation. We say that $\mathcal{C}$ is an accepting computation (respectively, rejecting computation) if the object `yes` (respectively, `no`) appears in the environment associated to the corresponding halting configuration of $\mathcal{C}$.

Definition 4.1 We say that a decision problem $X = (I_X, \theta_X)$ is solvable in polynomial time by a family $\Pi = \{\Pi(n) : n \in \mathbb{N}\}$ of recognizer tissue P systems with cell division if the following holds:

- The family Π 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 Π 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 Π 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 Π 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 have imposed to 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 $G = (V, E)$ is a function $f : V \to \{1, \ldots, k\}$. We say that G is k–colorable if there exists a k–coloring, f, such that $f(u) \neq f(v)$ for every edge $\{u, v\} \in E$ (the k–coloring f is *valid*). If we represent the numbers $1, 2, \ldots, n$ as colors, then adjacent vertices must have different colors by a valid k–coloring..

The k–coloring problem is the following: *given an undirected graph G, decide whether or not G is k-colorable*; that is, if there exists a valid 3–coloring of G.

This problem is related to the famous Four Color Conjecture (proved by Appel and Haken [3,4]). It is a special case of the problem of k-colorability of a graph, in which the range of C is $\{1, \ldots, k\}$ with k being specified as part of the instance. The **NP**-completeness of the 3–coloring problem was proved by Stockmeyer [28] (see [9]).

Next, we will see that the 3–coloring problem can be solved in linear time by a family of recognizer tissue P systems with cell division.

Let us consider a graph $G = (V, E)$, where $V = \{A_i : 1 \leq i \leq n\}$ is the set of vertices and $E \subseteq \{\{A_i, A_j\} : 1 \leq i < j \leq n\}$ is the set of edges. If $\{A_i, A_j\} \in E$ then we denote $A_{ij} = \{A_i, A_j\}$.

We will address the resolution via a brute force algorithm, in the framework of recognizer tissue P systems with cell division, which consists in the following phases:

- *Generation Stage*: The initial cell, labelled by 2, is divided into two new cells; and the division are iterated until we have all possible candidate solutions to the problem (one solution for each membrane). Simultaneously, in the membrane labelled by 1 a counter evolves, and it 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*: Once we obtain some objects R_{ij}, G_{ij}, B_{ij} in cells labelled by 2, we check if there exists a pair of adjacent vertices with the same color in the corresponding candidate solution.
- *Output Stage*: The system sends to the environment the right answer according

to the results of the previous stage.

Next, we provide a linear-time solution for the 3–coloring problem by a family of recognizer tissue P systems with cell division.

Let us recall that the function $\langle n, m\rangle = ((n+m)(n+m+1)/2)+n$ is primitive recursive and bijective from $\mathbb{N}^2$ onto $\mathbb{N}$. Also, the inverse function of h is polynomial.

For each $n, m \in \mathbb{N}$, we will 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

$$\{A_i, R_i, T_i, B_i, G_i, \overline{R}_i, \overline{B}_i, \overline{G}_i \,:\, 1 \leq i \leq n\} \cup$$

$$\{a_i \,:\, 1 \leq i \leq 3n + \lceil log_2\ m\rceil + 11\} \ \cup\ \{c_i \,:\, 1 \leq i \leq 2n+1\} \cup$$

$$\{d_i \,:\, 1 \leq i \leq \lceil log_2\ m\rceil + 1\} \ \cup \{f_i \,:\, 2 \leq i \leq n + \lceil log_2\ m\rceil + 6\} \cup$$

$$\{A_{ij}, P_{ij}, \overline{P}_{ij}, R_{ij}, B_{ij}, G_{ij} \,:\, 1 \leq i < j \leq n\} \cup \{b, D, E, e, T, S, N, \flat, \texttt{yes}, \texttt{no}\}$$

- $\Sigma(n) = \{A_{ij} \,:\, 1 \leq i < j \leq n\}$
- $w_1 = \{\{a_1, b, c_1, \texttt{yes}, \texttt{no}\}\}$
- $w_2(n) = \{\{D, A_1, \ldots, A_n\}\}$
- $\mathcal{R}(\langle n, m\rangle)$ is the set of rules:

(i) **Division rules:**

$r_{1,i} \equiv [A_i]_2 \rightarrow [R_i]_2[T_i]_2$ for $i = 1, \ldots, n$

$r_{2,i} \equiv [T_i]_2 \rightarrow [B_i]_2[G_i]_2$ for $i = 1, \ldots, n$

(ii) **Communication rules:**

$r_{3,i} \equiv (1, a_i/a_{i+1}, 0)$ for $i = 1, \ldots, 2n + \lceil log_2\ m\rceil + 10$

$r_{4,i} \equiv (1, c_i/c_{i+1}^2, 0)$ for $i = 1, \ldots, 2n$

$r_5 \equiv (1, c_{2n+1}/D, 2)$

$r_6 \equiv (2, c_{2n+1}/d_1 E, 0)$

$r_{7,i} \equiv (2, d_i/d_{i+1}^2, 0)$ for $i = 1, \ldots, \lceil log_2\ m\rceil$

$r_8 \equiv (2, E/e\ f_2, 0)$

$r_{9,i} \equiv (2, f_i/f_{i+1}, 0)$ for $i = 2, \ldots, n + \lceil log_2\ m\rceil + 5$

$r_{10,ij} \equiv (2, d_{\lceil log_2\ m\rceil+1} A_{ij}/P_{ij}, 0)$ for $1 \leq i < j \leq n$

$r_{11,ij} \equiv (2, P_{ij}/R_{ij}\overline{P}_{ij}, 0)$ for $1 \leq i < j \leq n$

$r_{12,ij} \equiv (2, \overline{P}_{ij}/B_{ij}G_{ij}, 0)$ for $1 \leq i < j \leq n$

$r_{13,ij} \equiv (2, R_iR_{ij}/R_i\overline{R}_j, 0)$ for $1 \leq i < j \leq n$

$r_{14,ij} \equiv (2, B_iB_{ij}/B_i\overline{B}_j, 0)$ for $1 \leq i < j \leq n$

$r_{15,ij} \equiv (2, G_iG_{ij}/G_i\overline{G}_j, 0)$ for $1 \leq i < j \leq n$

$r_{16,j} \equiv (2, \overline{R}_jR_j/\flat, 0)$ for $1 \leq j \leq n$

$r_{17,j} \equiv (2, \overline{B}_jB_j/\flat, 0)$ for $1 \leq j \leq n$

$r_{18,j} \equiv (2, \overline{G}_jG_j/\flat, 0)$ for $1 \leq j \leq n$

$r_{19} \equiv (2, e\,\flat/\lambda, 0)$

$r_{20} \equiv (2, e\ f_{n+\lceil log_2\ m\rceil+6}/T, 0)$

$r_{21} \equiv (2, T/\lambda, 1)$
$r_{22} \equiv (1, bT/S, 0)$
$r_{23} \equiv (1, S\ \texttt{yes}/\lambda, 0)$
$r_{24} \equiv (1, b\ a_{3n+\lceil log_2\ m\rceil+11}/N, 0)$
$r_{25} \equiv (1, N\ \texttt{no}/\lambda, 0)$

- $\mathcal{E}(\langle n, m\rangle) = \Gamma(\langle n, m\rangle) - \{\texttt{yes}, \texttt{no}\}$
- $i_{in} = 2$ 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 Π 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 codification of the instance will be the multiset $cod(u) = E$.

Next we describe informally 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 answer 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 codifies a vertex of the graph) we get (in two steps) three cells labelled by 2, each of them encoding one of the three colors 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. In this moment, the content of the cell labelled by 1 is $\{\{a_{2n+1}, c_{2n+1}^{4^n}, b, \texttt{yes}, \texttt{no}\}\}$, and there are 3^n cells each of them containing the object D, the objects A_{ij}, and a function from V to $\{1, 2, 3\}$ (we will identify the colors by *red*, R, *green*, G, and *blue*, B).

In the step $2n + 1$, 3^n copies of c_{2n+1} in the cell 1 is traded by 3^n objects D that appears one in 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 the cell labelled by 2, the communication starts.

At the beginning of the process, we pay attention to the counters d and f. 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 $2n + \lceil log_2\ m\rceil + 2$–th step (for the sake of simplicity, we will denote $\gamma = 2n + \lceil log_2\ m\rceil + 2$). The object f will be useful in order 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 apllying 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.

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

Taking into account that in the worst case each node needs to check $\max\{degree(u) : u \in V\} \leq n$ adjacent nodes, then after $\gamma + n + 3$ steps no more rules of the types $r_{13,ij}$, $r_{14,ij}$, and $r_{15,ij}$ can be applied. The checking stage will finish at the step $\gamma + n + 5$, because 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 step $\gamma + n + 6$.

- *Affirmative answer*: If there exists a valid 3–coloring of the graph, then in some cell labelled by 2 from the configuration $C_{\gamma+n+5}$ we have the object e and the object $f_{\lceil log_2\ m\rceil+n+6}$. 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 `yes` is sent out to the environment by the application of the rule r_{23} in the step $\gamma + n + 9 = 3n + \lceil log_2\ m\rceil + 11$. 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 $C_{\gamma+n+5}$. So, in these cells no rules can be applied anymore. In the next three steps, only the counter a in cell 1 evolves. Hence, in this cell from the configuration $C_{\gamma+n+8}$ we have the objects b and $a_{3n+\lceil log_2\ m\rceil+11}$. Next, by applying the rule r_{24} we get from the environment an object N, and in the following step an object `no` is sent out to the environment (because we also have an object `no` in cell 1). The computation finishes in the step $\gamma + n + 10 = 3n + \lceil log_2\ m\rceil + 12$.

5.2 Necessary Resources

The presented family of tissue P systems that solves the 3–coloring problem is polynomially uniform by Turing Machines. It can be observed that the definition of the family is done in a recursive manner from a given instance, in particular from the constants n and m. Furthermore the necessary resources to build the tissue P system $\Pi(\langle n, m\rangle)$ are:

- Size of the alphabet: $3n^2 + 11n + 3\lceil log_2\ m\rceil + 28 \in \theta(n^2 + \lceil log_2\ m\rceil)$.
- Initial number of cells: $2 \in \theta(1)$.
- Initial number of objects: $n + m + 6 \in theta(n + m)$.
- Number of rules: $3n^2 + 6n + 3\lceil log_2\ m\rceil + 24 \in \theta(n^2 + \lceil log_2\ m\rceil)$.
- Maximal length of a rule: 4.

Hence, there exists a deterministic Turing machine working in linear time that constructs the tissue P system $\Pi(\langle n, m\rangle)$ from $\langle n, m\rangle \in \mathbb{N}$.

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 5.1 *3–coloring* $\in \mathbf{PMC}_{TD}$.

As a consequence of this result we have:

Theorem 5.2 $\mathbf{NP} \cup \mathbf{co} - \mathbf{NP} \subseteq \mathbf{PMC}_{TD}$.

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 Conclusions 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.

In this paper we propose 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 *hamiltoniam path* problem, etc. Moreover, the structure of the solution described can be also adapted for solving computationally hard numerical problems.

Recently, a new kind P system model (called spiking neural P systems) based on the idea of spiking neurons has been presented (see, for example, [11]). The motivation of this model coming from two directions: the attempt of membrane computing to pass from cell-like architectures to tissue-like or neural-like architectures (see [23], [19]), 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*) [15].

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 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 excelence TIC-581 of the Junta de Andalucía.

References

[1] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. and Walter, P. *The Molecular Biology of the Cell*, Fourth Edition, Garland Publ. Inc., London (2002).

[2] Alhazov, A., Freund, R. and Oswald, M. Tissue P Systems with Antiport Rules ans Small Numbers of Symbols and Cells. *Lecture Notes in Computer Science* **3572**, (2005), 100–111.

[3] Appel, K. and Haken, W. Every planar map is 4-colorable - 1: Discharging. *Illinois Journal of Mathematics* **21**, (1977), 429–490.

[4] Appel, K. and Haken, W. Every planar map is 4-colorable - 2: Reducibility. *Illinois Journal of Mathematics* **21**, (1977), 491–567.

[5] Bernardini, F. and Gheorghe, M. 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. *Lecture Notes in Computer Science* **3082**, (2005), 257–278.

[7] Freund, R., Păun, Gh. and Pérez-Jiménez, M.J. Tissue P Systems with channel states. *Theoretical Computer Science* **330**, (2005), 101–116.

[8] Frisco, P. and Hoogeboom, H.J. Simulating counter automata by P systems with symport/antiport. *Lecture Notes in Computer Science* **2597**, (2003), 288–301.

[9] Garey, M.R. and Johnson, D.S. *Computers and Intractability A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, (1979).

[10] Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J. and Romero-Campero, F.J. A linear solution for QSAT with Membrane Creation. *Lecture Notes in Computer Science* **3850**, (2006), 241–252.

[11] Ionescu, M., Păun, Gh. and Yokomori, T. Spiking neural P systems, *Fundamenta Informaticae*, 71, 2-3 (2006), 279–308.

[12] Krishna, S.N., Lakshmanan K. and Rama, R. Tissue P Systems with Contextual and Rewriting Rules. *Lecture Notes in Computer Science* **2597**, (2003), 339–351.

[13] Lakshmanan K. and Rama, R. 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), 304–318.

[14] Luisi, P.L. The Chemical Implementation of Autopoiesis. In G.R. Fleishaker et al. (eds.) *Self-Production of Supramolecular Structures*, Kluwer, Dordrecht, (1994).

[15] W. Maass, C. Bishop, eds.: *Pulsed Neural Networks*, MIT Press, Cambridge, (1999).

[16] Martín Vide, C. Pazos, J. Păun, Gh. and Rodríguez Patón, A. A New Class of Symbolic Abstract Neural Nets: Tissue P Systems. *Lecture Notes in Computer Science* **2387**, (2002), 290–299.

[17] Martín Vide, C. Pazos, J. Păun, Gh. and Rodríguez Patón, A. Tissue P systems. *Theoretical Computer Science*, **296**, (2003), 295–326.

[18] Păun, Gh. Computing with membranes. *Journal of Computer and System Sciences*, **61**, 1, (2000), 108–143.

[19] Păun, Gh. *Membrane Computing. An Introduction*. Springer–Verlag, Berlin, (2002).

[20] Păun, A. and Păun, Gh. The power of communication: P systems with symport/antiport. *New Generation Computing*, **20**, 3, (2002), 295–305.

[21] Păun, Gh. and Pérez-Jiménez, M.J. Recent computing models inspired from biology: DNA and membrane computing. *Theoria*, **18**, 46, (2003), 72–84.

[22] Păun, Gh., Pérez-Jiménez, M.J. and Riscos-Núñez, A. 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), 380–386.

[23] Gh. Păun, Y. Sakakibara, T. Yokomori: P systems on graphs of restricted forms. *Publicationes Mathematicae Debrecen*, 60 (2002), 635–660.

[24] Pérez-Jiménez, M.J., Romero-Jiménez, A. and Sancho-Caparrini, F. *Teoría de la Complejidad en Modelos de Computación Celular con Membranas*. Editorial Kronos, Sevilla, (2002).

[25] Pérez-Jiménez, M.J., Romero-Jiménez, A. and Sancho-Caparrini, F. The P versus NP problem through cellular computing with membranes. *Lecture Notes in Computer Science* **2950**, (2004), 338–352.

[26] Pérez-Jiménez, M.J., Romero-Jiménez, A. and Sancho-Caparrini, F. 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), 284–294.

[27] Prakash, V.J. 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), 356–364.

[28] Stockmeyer, L.J. Planar 3-colorability is NP-complete. *SIGACT News* **5**, 3, (1973), 19–25.

[29] Zandron, C., Ferreti, C. and Mauri, G. Solving NP-Complete Problems Using P Systems with Active Membranes. In I. Antoniou, C.S. Calude and M.J. Dinneen (eds.). *Unconventional Models of Computation, UMC'2K*, Springer-Verlag, (2000), 289–301.

[30] ISI web page http://esi-topics.com/erf/october2003.html

[31] P systems web page http://psystems.disco.unimib.it/