Solving the 3-COL problem by using tissue P systems without environment and proteins on cells

Daniel Díaz-Pernil[a], Hepzibah A. Christinal[b], Miguel A. Gutiérrez-Naranjo[c,*]

[a] CATAM Research Group - Department of Applied Mathematics I, University of Seville, Spain
[b] Karunya University, Coimbatore, Tamilnadu, India
[c] Department of Computer Science and Artificial Intelligence, University of Seville, Spain

ARTICLE INFO

Article history:
Received 18 October 2016
Revised 7 November 2017
Accepted 14 November 2017
Available online 15 November 2017

Keywords:
Membrane computing
P systems
3-COL problem

ABSTRACT

The 3-COL problem consists on deciding if the regions of a map can be coloured with only three colors bearing in mind that two adjacent regions must be coloured with different colors. It is a **NP** problem and it has been previously used in complexity studies in membrane computing to check the ability of a model for solving problems of such complexity class. Recently, tissue P systems with proteins on cells have been presented and its ability to solve **NP** problems has been proved, but it remained as an open question to know if such model was still able to solve such problems if the environment was removed. In this paper, we provide an affirmative answer to this question by showing a uniform family of tissue P systems without environment and with proteins on cells which solves the 3-COL problem in linear time.

1. Introduction

The **P** versus **NP** problem is one of the most important unsolved problem in computer science and it was chosen as one of the seven Millennium Prize Problems [8]. The precise statement of the problem was introduced in 1971 by Cook [4], although it was essentially mentioned in a personal communication between K. Gödel and J. von Neumann [7].

Whereas the main question is unsolved (i.e., to decide if **P** and **NP** are or not the same complexity class), many efforts have been oriented in the last years in order to find *frontiers of tractability*, i.e., to identify some features of the computational models such that the corresponding device is able to solve or not **NP** problems depending if it is endowed or not with such feature.

Membrane computing [21] is a bio-inspired research area where many efforts have been addressed in order to discover new frontiers able to shed a new light on the **P** vs. **NP** problem. In this research area, the computational devices are called *P systems*. They consist on encapsulated pieces of information represented as multisets of objects placed in membranes and a set of rules which allows modify the multisets or even modify the membrane structure. In membrane computing, there exists an extensive literature devoted to this issue (see [27] and the references therein) and the present paper is a novel contribution in such research area. Recently, many other research lines in membrane computing have been open, see e.g., [11,22,23,28,33,34].

* Corresponding author.
E-mail addresses: sbdani@us.es (D. Díaz-Pernil), magutier@us.es (M.A. Gutiérrez-Naranjo).

https://doi.org/10.1016/j.ins.2017.11.022

We consider here a variant of one of the most popular P systems architectures: tissue P systems. Such model was firstly presented in [13,14] by placing the cells in a general graph instead on a tree-like graph as in the cell-like model. Under the hypothesis **P**≠**NP**, Zandron et al. [35] established the limitations of P systems that do not use membrane division concerning the efficient solution of **NP**-complete problems. From this premise, Păun et al. presented in [19] the model of tissue P systems endowed with cell division, able to solve **NP**-problems. Since then, many other variants have been presented, e.g., [5,9,10,16,17].

Tissue P systems with protein on cells have been introduced by Song et al. [29]. Previously, tissue P systems with proteins on membranes had been presented [20] and many of their properties have been explored (see, e.g., [18,30,31]). Nonetheless, the model of tissue P systems with protein on cells is quite different to the model with proteins on membranes: In the first one, proteins can move with multisets of objects but they cannot change. In the model with proteins on membranes, they can be changed, but they cannot move between membranes.

The ability of tissue P system with proteins on cells for solving **NP** problems has been proved [15,20], but it is an open question to know if after dropping some of the features, the model is still able to solve **NP** problems. In this paper[1], we prove that the model of tissue P system with proteins on cells can solve **NP** problems if the environment is not considered.

The environment in tissue P systems has a singular feature which makes it different to any other region: based on a biological inspiration, cells can take from the environment the necessary resources for any computation in a similar way that a biological cell can take as many oxygen molecules from the atmosphere (or the corresponding environment) as it needs. This means that the number of objects in the environment is not important and the designer does not need to take care of it. Avoiding the environment is a strong restriction, since all the resources are inside the cells from the beginning of the computation and nothing is taken from outside. The importance of the environment in other membrane computing models has been previously discussed in the literature (see, e.g. [3,12,25,26]). In this paper, we provide a uniform family of P systems with proteins on cells without environment which solves the 3-COL problem in linear time and hence, we prove that such systems are able to solve **NP** problems even if the environment is not considered.

The paper is organized as follows: Next, a formal description of the P system model used in this paper is given and some basics on recognizer P systems are recalled. In Section 3, we present the uniform family of P systems which solve the 3-COL problem in linear time and prove that **NP** problems can be solved in this model in polynomial time. Finally, the paper ends with some conclusions and open research lines.

2. Formal framework

Tissue P systems[2] with proteins on cells and cell division were introduced in [29]. In the same paper, the definition of *recognizer* tissue P systems [19] was presented in this framework. We adapt these definitions to the case where the environment is not considered.

Definition 1. A tissue P system without environment, with protein on cells and cell division of degree $q \geq 1$ is a tuple of the form

$$\Pi = (\Gamma, P, M_1/p_1, \ldots, M_q/p_q, \mathcal{R}, i_{in}, i_{out}),$$

where:

- Γ and P are finite non–empty alphabets such that $\Gamma \cap P = \emptyset$; Γ is the working alphabet and P is the set of proteins;
- $\mathcal{M}_i$ are finite multisets over Γ, $1 \leq i \leq q$;
- p_i are elements from P, $1 \leq i \leq q$;
- $\mathcal{R}$ is a finite set of rules of the following types:
  - *Communication rules:* $(i, (p_k, u)/(p_l, v), j)$, for $i, j \in \{1, \ldots, q\}$, $i \neq j$, $p_k, p_l \in P$, $u, v \in \Gamma^*$. The length of a communication rule is the total number of objects and proteins involved in that rule.
  - *Division rules:* $[p_j|a]_i \rightarrow [p_k|b]_i\ [p_l|c]_i$ for $i \in \{1, \ldots, q\}$, $p_j, p_k, p_l \in P$, $a, b, c \in \Gamma$, $i \neq i_{out}$
- $i_{in}, i_{out} \in \{1, \ldots, q\}$.

A tissue P system without environment, with protein on cells and cell division can be viewed as a set of q cells, labelled by $\{1, \ldots, q\}$ such that $M_1, \ldots, M_q$ represent the finite multisets of objects initially placed in the q cells of the system and $p_1, \ldots, p_q$ represent one and only one copy of protein initially placed on the q cells of the system; i_{in} is the cell where the input is placed in the initial configuration; and i_{out} represents a distinguished cell which will encode the output of the system. A configuration of the P system at any instant is described by all multisets of objects over Γ associated with all the cells present in the system and the proteins presented on them. The initial configuration is $(M_1/p_1, \ldots, M_q/p_q)$. A communication rule of type $(i, (p_k, u)/(p_l, v), j)$ is applicable to a configuration at an instant if cell i contains the protein p_k and the multiset u of objects, cell j contains the protein p_l and the multiset v of objects (multisets u, v may be empty; the empty multiset will be denoted by the symbol λ). When applying such a rule, under the control of the proteins p_k on cell

[1] A preliminary version of this paper will appear in the Proceedings of the 14th Brainstorming Week on Membrane Computing (BWMC2016).

[2] The reader is supposed to be familiar with basic membrane computing concepts, see [21].

i and p_l on cell j, both the protein p_k and the multiset u of objects are sent from cell i to cell j, and simultaneously, the protein p_l and the multiset v of objects are sent from cell j to cell i.

A division rule $[p_j|a]_i \to [p_k|b]_i\ [p_l|c]_i$ is applicable to a configuration if cell i contains the protein p_j and the object a. When applying such a rule, under the influence of protein p_j and the object a in cell i, the cell is divided into two cells with the same label; in the first copy of the cell the protein p_j is replaced by p_k and the object a is replaced by b, in the second copy of the cell the protein p_j is replaced by p_l and the object a is replaced by c; all the remaining objects in the original cell are replicated and distributed in each of the new cells.

Rules are used in a maximally parallel way: at each step, all cells which can evolve must evolve and a maximal multiset of rules is applied (no further rule can be added being applicable). Let us remark that only one protein is placed in each cell and therefore only one rule can be applied to a cell in a computation step, regardless if it is a communication one or a division one.

2.1. Recognizer tissue P systems with protein on cells and cell division

The notion of recognizer P system is general enough to cover many P system variants. Roughly speaking, a recognizer P system is a P system which takes some information as *input* and outputs an object *yes* or *no* which can be considered as a *decision* on the input. Of course, some other conditions are imposed, but the general framework does not depend on the use of proteins or the existence of environment.

Next, we recall the main notions related to the theory of recognizer P systems, which fit into the presented model in a natural way. For a detailed description see, e.g., [24,27]. A decision problem X is a pair (I_X, θ_X) such that 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 general, in a *P system with input and output* of any P systems variant, we consider a working alphabet Γ, with q membranes labelled by $1, \ldots, q$, and initial multisets $\mathcal{M}_1, \ldots, \mathcal{M}_q$ associated with them; Σ, which is an (input) alphabet strictly contained in Γ; the initial multisets are over $\Gamma - \Sigma$; and i_{in}, i_{out} are the labels of two distinguished membranes (input and output). Let Γ be the working alphabet of Π, μ its membrane structure, and $\mathcal{M}_1, \ldots, \mathcal{M}_p$ the initial multisets[3] of Π. Let m be a multiset over Σ encoding the input. The multisets of objects placed into the membranes in the initial configuration are $\mathcal{M}_1, \ldots, \mathcal{M}_{i_{in}} \cup m, \ldots, \mathcal{M}_q$.

A *recognizer P system* is a P system with input and output such that:

- The working alphabet contains two distinguished elements *yes, no*.
- All its 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 output region (denoted with label i_{out}), and only in the last step of the computation. We say that $\mathcal{C}$ is an accepting computation (respectively, rejecting computation) if the object *yes* (respectively, *no*) appears in the output region associated to the corresponding halting configuration of $\mathcal{C}$.

A decision problem X can be solved in a polynomially uniform way by a family $\mathbf{\Pi} = \{\Pi(n)\}_{n\in\mathbb{N}}$ of P systems of type $\mathcal{F}$ if the following holds:

- There is a deterministic Turing machine M such that, for every $n \in \mathbb{N}$, starting M with the unary representation of n on its input tape, it constructs the P system $\Pi(n)$ in polynomial time in n.
- There is a deterministic Turing machine N that started with an instance $I \in I_X$ with size n on its input tape, it computes a multiset w_I (called the *encoding of* I) over the input alphabet of $\Pi(n)$ in polynomial time in n.
- For every instance $I \in I_X$ with size n, starting $\Pi(n)$ with w_I in its input membrane, every computation of $\Pi(n)$ halts and sends out to the environment *yes* if and only if I is a positive instance of X.

According to the standard notation, $\widehat{\mathbf{TPDC}}(k)$ denotes the class of recognizer tissue P systems without environment with protein on cells and communication rules of length at most k and $\mathbf{PMC}_{\widehat{TPDC}(k)}$ the set of all decision problems which can be solved by means of such class. This class is closed under polynomial time reduction and under complement.

3. The 3-COL 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 $\{1, \ldots, k\}$ 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*).

In particular, when $k = 3$, we have the well-known 3–coloring problem: *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 will 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

[3] In the case of P systems with proteins, the initial description also includes the corresponding proteins. The occurrence of such proteins does not have influence in the definitions of recognizer P systems.

the famous Four Color Conjecture (proved by Appel and Haken [1,2]). The **NP**-completeness of the 3–coloring problem was proved by Stockmeyer [32] (see [6]).

Next, we will prove that the 3–coloring problem can be solved in a linear time by a family of recognizer tissue P systems without environment and with proteins on cells. The resolution will be addressed via a brute force algorithm, which consists in the following stages:

- *Generation stage*: All the possible 3–coloring are generated, each of them placed in a different cell. This is done by using division rules and an exponential amount of cells is obtained in linear time.
- *Checking stage*: If a generated 3–coloring has two objects K_i and K_j ($K \in \{R, G, B\}$) and the graph has an edge A_{ij} linking the nodes i and j, this coloring is not valid. In this case, cells encoding non-valid colorings change their proteins. If a cell encoding a colouring keeps its protein after this stage, it means that it is a valid colouring. This stage takes only one step.
- *Output stage*: It suffices that one of the possible coloring satisfies the conditions in order to have a positive answer. If such coloring exists, a distinguished protein will be sent to the appropriate cell. We can control via a counter the number of steps for it. If such protein occurs in the right cell at the right moment, the system sends `yes` to the output cell. If such step is reached and the protein has not been released, an object `no` is sent to the output cell.

Each of the P systems of the uniform family $\mathbf{\Pi} = \{\Pi_n\}_{n\in\mathbb{N}}$ described below depends only on one parameter n which represents the number of nodes of the graph. Each of these Π_n is supplied with the encoding of a concrete instance of a graph with n vertices in order to start the computation. The graph will be encoded by using an *input alphabet* $\Sigma = \{A_{ij} : 1 \leq i < j \leq n\}$, and an object A_{ij} will belong to the *input multiset* if and only if there is an edge in the graph linking the nodes i and j. For the sake of simplicity we drop the subscript in Π_n. Formally, for each $n \in \mathbb{N}$, the tissue P system is defined as

$$\Pi = (\Gamma, P, \Sigma, \mathcal{M}_1/p_1, \mathcal{M}_2/p_2, \mathcal{M}_3/p_3, \mathcal{M}_4/p_4, \mathcal{M}_5/p_5, \mathcal{M}_6/p_6, \mathcal{R}, i_{in}, i_{out}),$$

- $\Gamma = \Sigma \cup \{A_i, R_i, G_i, B_i, U_i, V_i : 1 \leq i \leq n\}$
  $\cup\{a_i : 0 \leq i \leq 2n+1\}$
  $\cup\{b_i : 0 \leq i \leq 2n+2\}$
  $\cup\{T, \texttt{yes}, \texttt{no}\}$
- $\Sigma = \{A_{ij} : 1 \leq i < j \leq n\}$
- $P = \{p_{i,j} : i \in \{1,2\}\ j \in \{1,\ldots,2n+1\}$
  $\cup\{q_{i,j} : i \in \{1,2\}\ j \in \{1,\ldots,2n\}$
  $\cup\{p_0\}$
- $\mathcal{M}_1 = \{A_1, \ldots, A_n\}$ with the initial protein $p_{1,1}$ in cell 1;
- $\mathcal{M}_2 = \{A_1, \ldots, A_n\}$ with the initial protein $p_{2,1}$ in cell 2;
- $\mathcal{M}_3 = \{a_0\}$ with the initial protein p_0 in cell 3;
- $\mathcal{M}_4 = \{b_0, \texttt{yes}, \texttt{no}\}$ with the initial protein p_0 in cell 4;
- $\mathcal{M}_5 = \{a_1, \ldots, a_{2n+1}\}$ with the initial protein p_0 in cell 5;
- $\mathcal{M}_6 = \{b_1, \ldots, b_{2n+2}\}$ with the initial protein p_0 in cell 6;
- $\mathcal{R}$ is the following set of rules:
  1. *Division rules:* For $i \in \{1, 2\}$ and $j \in \{1, \ldots, n\}$
     $r_{1,i,j} \equiv [p_{i,j}|A_j]_i \to [q_{i,j}|U_j]_i\ [q_{i,j}|V_j]_i$
     $r_{2,i,j} \equiv [q_{i,j}|U_j]_i \to [p_{i,j+1}|R_j]_i\ [p_{i,j+1}|G_j]_i$
     $r_{3,i,j} \equiv [q_{i,j}|V_j]_i \to [p_{i,j+1}|B_j]_i\ [p_{i,j+1}|T]_i$
  2. *Communication rules:*
     $r_{4,i,j,K} \equiv (1, (p_{1,2n+1}, A_{ij})/(p_{2,2n+1}, K_i K_j), 2)$ for $i, j \in \{1, \ldots, n\}$, $i < j$, $K \in \{R, G, B\}$
     $r_{5,i} \equiv (3, (p_0, a_i)/(p_0, a_{i+1}), 5)$ for $i = \{0, \ldots, 2n\}$
     $r_{6,i} \equiv (4, (p_0, b_i)/(p_0, b_{i+1}), 6)$ for $i = \{0, \ldots, 2n+1\}$
     $r_7 \equiv (2, (p_{2,2n+1}, \lambda)/(p_0, a_{2n+1}), 3)$
     $r_8 \equiv (4, (p_0, b_{2n+2}\,\texttt{yes})/(p_{2,2n+1}, \lambda), 3)$
     $r_9 \equiv (4, (p_0, b_{2n+2}\,\texttt{no})/(p_0, \lambda), 3)$
- $i_{in} = 1$, is the input cell
- $i_{out} = 3$, is the output cell

Fig. 1 shows a scheme of the initial configuration and the communication rules. The presented uniform family $\mathbf{\Pi} = \{\Pi_n\}_{n\in\mathbb{N}}$ is the key point for proving the main result of this paper

Theorem 1. *3-COL* $\in$ $\boldsymbol{PMC}_{\widehat{TPDC(5)}}$

Before going on with an outline of the formal proof. We provide some intuitions with an overview of the computation. Each P system of the family is deterministic and it exploits the parallelism intrinsic to membrane computing systems and the specific feature of tissue P system with proteins on cells which fix one and only one protein in each membrane.

From the initial configuration, four processes start:

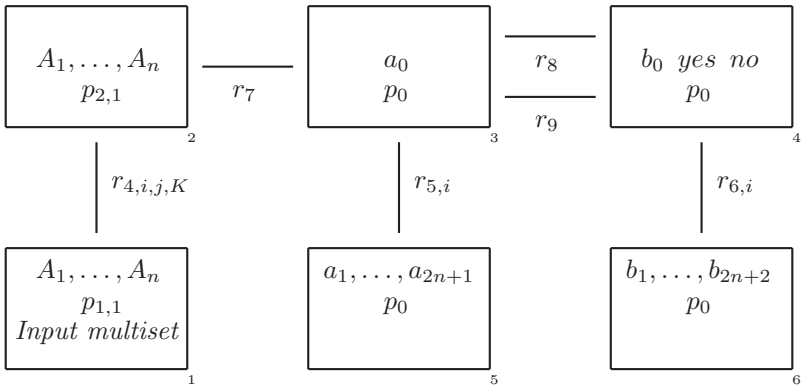

Fig. 1. Scheme of the initial configuration with the initial multisets and proteins. The communication rules are also marked.

1. Cell 1 is divided by the application of rules $r_{1,1,j}$, $r_{1,2,j}$ and $r_{1,3,j}$. The configuration $\mathbb{C}_{2n}$ has 2^{2n} membranes with label 1, each of them containing a copy of the input and a protein $p_{1,2n+1}$.
2. Cell 2 is divided by the application of rules $r_{2,1,j}$, $r_{2,2,j}$ and $r_{2,3,j}$ in parallel with the cell of label 1. The configuration $\mathbb{C}_{2n}$ has 2^n membranes with label 2 all of them with the protein $p_{2,2n+1}$. Some of these membrane contain one or more copies of the object T. Each of the remaining 3^n membranes contain a different 3-coloring, i.e., a multiset of objects $C_1C_2 \ldots C_n$ with $C_i \in \{R_i, G_i, B_i\}$.
3. Cell 3 interchanges one object a_i with cell 5 during the $2n$ first steps, so at $\mathbb{C}_{2n}$ it contains the protein p_0 and the object a_{2n}.
4. Analogously, cell 4 interchanges one object b_i with cell 6 during the $2n$ first steps, so at $\mathbb{C}_{2n}$ it contains the protein p_0 and the object b_{2n}.

At the configuration $\mathbb{C}_{2n}$, cells 1 contain the protein $p_{1,2n+1}$ and cells 2 contain the protein $p_{2,2n+1}$. If a cell 2 contain two objects K_iK_j with the same color ($K \in \{R, G, B\}$) and there exists an edge A_{ij} in the input, then the rule $r_{4,i,j,K}$ is applied and the corresponding cells interchange their proteins. Since there are enough cells with label 1, the following holds:

- If a cell 2 represent a valid coloring, then the rule $r_{4,i,j,K}$ is not applied and the cell has the protein $p_{2,2n+1}$ at the configuration $\mathbb{C}_{2n+1}$.
- Otherwise, if the coloring represented in the cell is not valid, then the rule $r_{4,i,j,K}$ is applied and the cell has the protein $p_{1,2n+1}$ at the configuration $\mathbb{C}_{2n+1}$.
- Moreover, at $\mathbb{C}_{2n+1}$, cell 3 has protein p_0 and an object a_{2n+1} and cell 4 has protein p_0 and and object b_{2n+1}

Let us recall that if there exists at least one valid coloring, then the answer to the 3-COL problem must be affirmative. Let us consider that there exist such valid coloring and then, at $\mathbb{C}_{2n+1}$ there exists (at least) one cell 2 with protein $p_{2,2n+1}$. In such case the rule 7 applied and at $\mathbb{C}_{2n+2}$ the cell 3 contains the protein $p_{2,2n+1}$. Otherwise, if none of the cells 2 has the protein $p_{2,2n+1}$, then the rule 7 is not applied and cell 3 has the protein p_0 at $\mathbb{C}_{2n+2}$. In such configuration the object b_{2n+2} has reached cell 4. Finally, depending on the protein p_0 or $p_{2,2n+1}$ in cell 3, rule 8 or rule 9 will be applied sending the right answer to cell 3. No more rules can be applied and $\mathbb{C}_{2n+3}$ is a halting configuration.

Outline of the proof The result can be obtained from the following partial results.

Lemma 1. *The family* $\mathbf{\Pi} = \{\Pi_n\}_{n\in\mathbb{N}}$ *is polynomially uniform*

This lemma can be proven with standard techniques of membrane computing complexity [27]. In order to prove it, it must checked that the initial resources depends polynomially on the parameter n. The amount of resources used in the construction of the P system Π_n can be summarized as follows: The working alphabet Γ is $O(n)$ with $10n + 8$ objects; the input alphabet is $O(n^2)$ with $\frac{n^2-n}{2}$ objects; the set of proteins is $O(n)$ with $6n + 3$ proteins; and the number of rules is $O(n^2)$ with $\frac{3}{2}n^2 + \frac{17}{2}n + 6$ rules. Moreover, all the computation halt after $2n + 3$ steps and the communication rules have length 5 at most. Therefore the main result of this paper holds.

Lemma 2. *Given* $r \in \{0, \ldots, 2n\}$, *there are* 2^r *cells with label 2 at the configuration* $\mathcal{C}_r$.

In the initial configuration $\mathcal{C}_0$, there is $2^0 = 1$ cell with label 1 and protein $p_{2,1}$. By application of the unique applicable rule, $r_{1,2,1}$, two cells with label 2 and protein $q_{2,1}$ are obtained at $\mathcal{C}_1$. Next, by application of rules $r_{2,2,1}$ and $r_{3,2,1}$, $2^2 = 4$ cells with label 2 and protein $p_{2,2}$ are obtained at $\mathcal{C}_2$. By iteration of the process, 2^{2n} cells with label 2 and protein $p_{2,2n+1}$ are obtained at $\mathcal{C}_{2n}$.

Lemma 3. *Given* $r \in \{1, \ldots, n\}$

- *The multiset of each cell with label 2 at the configuration $\mathcal{C}_{2r}$ is one of the following ones*

$$p_{2,r+1}, K_1, \ldots, K_r, A_{r+1}, \ldots, A_n$$

with $K_i \in \{R_i, G_i, B_i, T\}$ for $i \in \{1, \ldots, r\}$.

- *The multiset of each cell with label 2 at the configuration $\mathcal{C}_{2r-1}$ is one of the following ones*

$$q_{2,r}, K_1, \ldots, K_{r-1}, W_r, A_{r+1}, \ldots, A_n$$

with $W_r \in \{U_r, V_r\}$, $K_i \in \{R_i, G_i, B_i, T\}$ for $i \in \{1, \ldots, r-1\}$.

This lemma can be easily checked bearing in mind that the P system is deterministic and the evolution of the multiset of a generic cell with label 2 can be summarized as follows

$$\begin{array}{rll}
\mathcal{C}_\prime \equiv & p_{2,1} A_1 A_2 A_3 \ldots A_n & -- \ r_{1,2,1} \text{ is applied} \\
\mathcal{C}_1 \equiv & q_{2,1} W_1 A_2 A_3 \ldots A_n & -- \ r_{k,2,1},\ k \in \{2,3\} \text{ is applied} \\
\mathcal{C}_2 \equiv & p_{2,2} K_1 A_2 A_3 \ldots A_n & -- \ r_{1,2,2}, \text{ is applied} \\
\mathcal{C}_3 \equiv & q_{2,2} K_1 W_2 A_3 \ldots A_n & -- \ r_{k,2,2},\ k \in \{2,3\} \text{ is applied} \\
\ldots & \ldots & \\
\mathcal{C}_{2n} \equiv & p_{2,2n+1} K_1 K_2 K_3 \ldots K_n &
\end{array}$$

Let us recall that Lemmas 2 and 3 refer to cells with label 2. Analogous lemmas for cells with label 1 can be proved. The unique difference concerns to the codification of the input, which is placed in all the copies of cells with label 1 at $\mathcal{C}_{2n}$.

Lemma 4. *If there exists a valid colouring for the instance of the problem, then the P system halts and an object yes is placed in the output cell in the last step of computation. Otherwise, if there does not exist a valid colouring for the instance of the problem, then the P system halts and an object no is placed in the output cell in the last step of computation.*

The key features here are, on the one hand, the massive parallelism of the P system and, on the other hand, that each cell can be involved in the application of one rule at most. Since there are the same number, 2^{2n} cells of label 1 and 2 at the configuration $\mathcal{C}_{2n}$, all the cells 1 and 2 encoding a non valid colouring interchange their proteins. In this way, all cells of label 2 which contains a protein $p_{2,2n+1}$ at the configuration C_{2n+1} represents a valid colouring and all cells of label 2 which contains a protein $p_{1,2n+1}$ at the configuration C_{2n+1} represents a non-valid colouring. It may exist several valid colouring, but in this case, rule 7 is applied once. In $\mathcal{C}_{2n+2}$ there are two possibilities for the cell with label 3. If there exist at least one valid colouring, it contains a protein $p_{2,2n+1}$; otherwise, if all the colouring are not valid, it contains a protein p_0. Finally, in the next step rule 8 or rule 9 is applied and the corresponding answer *yes* or *not* is placed in the output cell. Let us notice that these rules cannot be applied before because they need to wait the occurrence of the object b_{2n+2} in the cell with label 4.

Corollary 1. $\boldsymbol{NP} \cup co\text{-}\boldsymbol{NP} \subseteq \boldsymbol{PMC}_{\widehat{TPDC(5)}}$

This result holds from the previous theorem and the closure under polynomial-time reduction and under complement of the complexity class.

4. Conclusions

Whereas the **P** vs. **NP** is unsolved, the search of new frontiers of tractability allows us to have a deeper knowledge of the problem. In the framework of membrane computing, and in natural computing in general, the use of bio-inspired features in such complexity studies sheds a new light on an old problem. In this paper, we present a new solution to the 3-COL problem with tissue P systems with proteins on cells and without environment which uses communication rules of length at most 5. By using environment, the solution for the SAT problem proposed in [29] uses communication rules of length at most 4. In [15], the proposed solution for the 3-COL problem also uses communication rules of length at most 4. Although both problems, SAT and 3-COL, are different, it remains open the question if it is possible to find a solution to a **NP** problem in the model of tissue P systems with proteins on cells by removing the environment and using communication rules of length at most 4. Further research can be done by considering new ways to generate exponential space.

References

[1] K. Appel, W. Haken, Every planar map is 4-colorable - 1: Discharging, Illinois J. Math. 21 (1977) 429–490.
[2] K. Appel, W. Haken, Every planar map is 4-colorable - 2: Reducibility, Illinois J. Math. 21 (1977) 491–567.
[3] H.A. Christinal, D. Díaz-Pernil, M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, Tissue-like p systems without environment, in: M.A. Martínez del Amor, Gh. Păun, I. Pérez Hurtado, A. Riscos-Núñez (Eds.), Eighth Brainstorming Week on Membrane Computing, Fénix Editora, Sevilla, Spain, 2010, pp. 53–64.
[4] S.A. Cook, The complexity of theorem-proving procedures, in: Proceedings of the Third Annual ACM Symposium on Theory of Computing, STOC '71, ACM, New York, NY, USA, 1971, pp. 151–158.
[5] R. Freund, Gh. Păun, M.J. Pérez-Jiménez, Tissue P systems with channel states, Theoretical Comput. Sci. 330 (2005) 101–116.
[6] M.R. Garey, D.S. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, New York, 1979.
[7] J. Hartmanis, Gödel, von Neumann and the P =? NP problem, in: G. Rozenberg, A. Salomaa (Eds.), Current Trends in Theoretical Computer Science - Essays and Tutorials, World Scientific Series in Computer Science, volume 40, World Scientific, 1993, pp. 445–450.
[8] A.M. Jaffe, The millennium grand challenge in mathematics, Notices Am. Math. Soc. 53 (2006) 652–660.

[9] S.N. Krishna, K. Lakshmanan, R. Rama, Tissue P systems with contextual and rewriting rules, in: Gh. Păun, G. Rozenberg, A. Salomaa, C. Zandron (Eds.), WMC-CdeA, Lecture Notes in Computer Science, volume 2597, Springer, Berlin Heidelberg, 2002, pp. 339–351.
[10] K. Lakshmanan, R. Rama, On the power of tissue P systems with insertion and deletion rules, in: A. Alhazov, C. Martín-Vide, Gh. Păun (Eds.), Preproceedings of the Workshop on Membrane Computing, Rovira i Virgili University, Tarragona, 2003, pp. 304–318.
[11] X. Liu, J. Suo, S.C.H. Leung, J. Liu, X. Zeng, The power of time-free tissue P systems: Attacking NP-complete problems, Neurocomputing 159 (2015) 151–156.
[12] L.F. Macías-Ramos, M.J. Pérez-Jiménez, A. Riscos-Núñez, M. Rius-Font, L. Valencia-Cabrera, The efficiency of tissue P systems with cell separation relies on the environment, in: E. Csuhaj-Varjú, M. Gheorghe, G. Rozenberg, A. Salomaa, G. Vaszil (Eds.), Membrane Computing - 13th International Conference, CMC 2012, Budapest, Hungary, August 28-31, 2012, Revised Selected Papers, Lecture Notes in Computer Science, volume 7762, Springer, 2012, pp. 243–256.
[13] C. Martín-Vide, Gh. Păun, J. Pazos, A. Rodríguez-Patón, Tissue P systems, Theoretical Comput. Sci. 296 (2003) 295–326.
[14] C. Martín-Vide, J. Pazos, Gh. Păun, A. Rodríguez-Patón, A new class of symbolic abstract neural nets: Tissue P systems, in: O.H. Ibarra, L. Zhang (Eds.), COCOON, Lecture Notes in Computer Science, volume 2387, Springer, Berlin Heidelberg, 2002, pp. 290–299.
[15] T. Mathu, H.A. Christinal, D. Díaz-Pernil, A uniform family of tissue P systems with protein on cells solving 3-coloring in linear time, in: C.S. Calude, M.J. Dinneen (Eds.), Unconventional Computation and Natural Computation - 14th International Conference, UCNC 2015, Auckland, New Zealand, August 30 - September 3, 2015, Proceedings, Lecture Notes in Computer Science, volume 9252, Springer, 2015, pp. 239–249.
[16] V. Pakash, On the power of tissue P systems working in the maximal-one mode, in: A. Alhazov, C. Martín-Vide, Gh. Păun (Eds.), Preproceedings of the Workshop on Membrane Computing, Rovira i Virgili University, Tarragona, 2003, pp. 356–364.
[17] L. Pan, M.J. Pérez-Jiménez, Computational complexity of tissue-like P systems, J. Complexity 26 (2010) 296–315.
[18] A. Păun, M. Păun, A. Rodríguez-Patón, M. Sidoroff, P systems with proteins on membranes: a survey, Int. J. Foundations Comput. Sci. 22 (2011) 39–53.
[19] Gh. Păun, M.J. Pérez-Jiménez, A. Riscos-Núñez, Tissue P systems with cell division, Int. J. Comput. Commun. Control 3 (2008) 295–303.
[20] A. Păun, B. Popa, P systems with proteins on membranes, Fundamenta Informaticae 72 (2006) 467–483.
[21] Gh. Păun, G. Rozenberg, A. Salomaa (Eds.), The Oxford Handbook of Membrane Computing, Oxford University Press, Oxford, England, 2010.
[22] H. Peng, J. Wang, M.J. Pérez-Jiménez, A. Riscos-Núñez, An unsupervised learning algorithm for membrane computing, Inf. Sci. 304 (2015) 80–91.
[23] H. Peng, J. Wang, M.J. Pérez-Jiménez, H. Wang, J. Shao, T. Wang, Fuzzy reasoning spiking neural P system for fault diagnosis, Inf. Sci. 235 (2013) 106–116.
[24] M.J. Pérez-Jiménez, An approach to computational complexity in membrane computing, in: G. Mauri, Gh. Păun, M.J. Pérez-Jiménez, G. Rozenberg, A. Salomaa (Eds.), Workshop on Membrane Computing, Lecture Notes in Computer Science, volume 3365, Springer, 2004, pp. 85–109.
[25] M.J. Pérez-Jiménez, A. Riscos-Núñez, M. Rius-Font, F.J. Romero-Campero, A polynomial alternative to unbounded environment for tissue P systems with cell division, Int. J. Comput. Math. 90 (2013) 760–775.
[26] M.J. Pérez-Jiménez, A. Riscos-Núñez, M. Rius-Font, L. Valencia-Cabrera, The relevance of the environment on the efficiency of tissue P systems, [1], 2013b, 308–321.
[27] M.J. Pérez-Jiménez, A. Riscos-Núñez, A. Romero-Jiménez, D. Woods, Complexity - membrane division, membrane creation, [22], 2010. 302–336.
[28] T. Song, L. Pan, Gh. Păun, Asynchronous spiking neural P systems with local synchronization, Inf. Sci. 219 (2013) 197–207.
[29] B. Song, L. Pan, M.J. Pérez-Jiménez, Tissue P systems with protein on cells, Fundamenta Informaticae 144 (2016) 77–107.
[30] P. Sosík, Active membranes, proteins on membranes, tissue P systems: Complexity-related issues and challenges, [1], (2013) 40–55.
[31] P. Sosík, A. Păun, A. Rodríguez-Patón, P systems with proteins on membranes characterize PSPACE, Theoretical Comput. Sci. 488 (2013) 78–95.
[32] L. Stockmeyer, Planar 3-colorability is NP-complete, SIGACT News 5 (1973) 19–25.
[33] J. Wang, P. Shi, H. Peng, Membrane computing model for IIR filter design, Inf. Sci. 329 (2016) 164–176.
[34] J. Wang, P. Shi, H. Peng, M.J. Pérez-Jiménez, T. Wang, Weighted fuzzy spiking neural P systems, IEEE Trans. Fuzzy Syst. 21 (2013) 209–220.
[35] C. Zandron, C. Ferretti, G. Mauri, Solving NP-complete problems using P systems with active membranes, in: I. Antoniou, C. Calude, M.J. Dinneen (Eds.), UMC, Springer, 2000, pp. 289–301.