Solving the 3-COL Problem by Using Tissue P Systems without Environment and Proteins on Cells

Daniel Díaz-Pernil[1], Hepzibah A. Christinal[2], Miguel A. Gutiérrez-Naranjo[3]

[1]Research Group on Computational Topology and Applied Mathematics
Department of Applied Mathematics - University of Sevilla, 41012, Spain
sbdani@us.es

[2]Karunya University, Coimbatore, Tamilnadu, India
christyhep@gmail.com

[3]Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla, 41012, Spain
magutier@us.es

Summary. 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 [9]. The precise statement of the problem was introduced in 1971 by Stephen Cook [5], although it was essentially mentioned in a personal communication between K. Gödel and J. von Neumann [8].

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.

In membrane computing there is an extensive literature devoted to this issue (see [21] and the references therein) and the present paper is a novel contribution in such research line. 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.* [29] established the limitations of P systems that do not use membrane division concerning the efficient solution of **NP**-complete problems. Under this premise, Gh. Păun *et al.* presented in [24] the model of tissue P systems with cell division, able to solve **NP**-problems. Since then, many other variants have been presented, e.g., [6, 10, 11, 16, 17].

Recently, tissue P systems with protein on cells have been introduced [25]. Previously, tissue P systems with proteins on membranes had been presented [22] and many of their properties have been explored (see, e.g., [23, 26, 27]). 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.

Tissue P system with proteins on cells is endowed with cell division and its ability for solving **NP** problems has been proved [15, 22], 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, we prove that the model of tissue P system with proteins on cells can solve **NP** problems if the environment is removed. The environment in tissue P systems has a singular characteristic 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 cell can take as many oxygen molecules from the atmosphere 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 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. [4, 12, 19, 20]). 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 of solving **NP** problems even the environment is dropped.

The paper is organized as follows: Next we give a formal description of the P system model used in this paper and recall some basics on recognizer P systems. In Section 3 we present the uniform family of P systems which solve the 3-COL problem in linear time and discuss the amount of resources needed. Finally, the paper ends with some conclusions.

2 Formal Framework

Tissue P systems with proteins on cells and cell division were introduced in [25]. In the same paper, the definition of *recognizer* tissue P systems [24] is 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, 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 all cells. 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 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 $[p_j|a]_i \rightarrow [p_k|b]_i \; [p_l|c]_i$ is applicable to a configuration at an instant 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). As usual in the variant of tissue P systems, this way of applying rules has only one restriction: when a cell is divided, the division rule is the only one which is applied to that cell at that step. The new cells resulting from division could participate in the interaction with other cells by means of communication rules at the next step (if they are not divided once again).

2.1 Recognizer Tissue P Systems with Protein on Cells and Cell Division

We recall the main notions related to the theory of recognizer P systems, which can be adapted to this model in a natural way. For a detailed description see, e.g., [18, 21]. 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 system 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 of Π. Let m be a multiset over Σ. The *initial configuration* of the P system is $(\mu, \mathcal{M}_1, \ldots, \mathcal{M}_{i_{in}} \cup m, \ldots, \mathcal{M}_q)$.

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 the famous Four Color Conjecture (proved by Appel and Haken [2, 3]). The **NP**-completeness of the 3–coloring problem was proved by Stockmeyer [28] (see [7]).

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. As usual, we will address the resolution 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. By using the division rules, an exponential amount of cells can be obtained in linear time. In parallel, the cell containing initially a copy of the description of the graph is also divided generating as many copies of the graph as 3-colorings.
- *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. Since the number of cells containing a copy of the description of the graph is large enough, the checking for all the colorings can be done in parallel by pairing cells encoding 3-colorings with cells encoding copies of the graph. 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 send 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

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

- $\Gamma = \Sigma \cup \{A_i, R_i, G_i, B_i, U_i, V_i : 1 \leq i \leq n\}$
  $\cup \; \{a_i : 0 \leq i \leq 2n+1\}$
  $\cup \; \{b_i : 0 \leq \; i \leq 2n+2\}$
  $\cup \; \{T, \texttt{yes}, \texttt{no}\}$
- $\Sigma = \{A_{ij} : 1 \leq i < j \leq n\}$
- $P = \{p_{i,j} : i \in \{1,2\} \, j \in \{1, \dots, 2n+1\}$
  $\cup \; \{q_{i,j} : i \in \{1,2\} \, j \in \{1, \dots, 2n\}$
  $\cup \; \{p_0\}$
- $\mathcal{M}_1 = \{A_1, \dots, A_n\}$ with the initial protein $p_{1,1}$ in cell 1;
- $\mathcal{M}_2 = \{A_1, \dots, A_n\}$ 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;
- $\mathcal{M}_5 = \{a_1, \dots, a_{2n+1}, b_1, \dots, b_{2n+2}\}$ with the initial protein p_0 in cell 5;
- $\mathcal{R}$ is the following set of rules:
  1. *Division rules:* For $i \in \{1,2\}$ and $j \in \{1, \dots, n\}$
     $r_{1,i,j} \equiv [p_{i,j}|A_j]_i \to [q_{i,j}|U_j]_i \; [q_{i,j}|V_j]_i$
     $r_{2,i,j} \equiv [q_{i,j}|U_j]_i \to [p_{i,j+1}|R_j]_i \; [p_{i,j+1}|G_j]_i$
     $r_{3,i,j} \equiv [q_{i,j}|V_j]_i \to [p_{i,j+1}|B_j]_i \; [p_{i,j+1}|T]_i$

  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 $i, j \in \{1, \dots, n\}$, $i < j$, $K \in \{R, G, B\}$
     $r_{5,i} \equiv (3, (p_0, a_i)/(p_0, a_{i+1}), 5)$ for $i = \{0, \dots, 2n\}$
     $r_{6,i} \equiv (4, (p_0, b_i)/(p_0, b_{i+1}), 5)$ for $i = \{0, \dots, 2n+1\}$
     $r_7 \equiv (2, (p_{2,2n+1}, \lambda)/(p_0, a_{2n+1}), 3)$
     $r_8 \equiv (4, (p_0, b_{2n+2} \, \texttt{yes})/(p_{2,2n+1}, \lambda), 3)$
     $r_9 \equiv (4, (p_0, b_{2n+2} \, \texttt{no})/(p_0, \lambda), 3)$
- $i_{in} = 1$, is the input cell
- $i_{out} = 3$, is the output cell

3.1 An Overview of the Computation

The system 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:

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^n 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 3-coloring, i.e., a multiset of objects $C_1C_2\dots C_n$ with $C \in \{R, G, B\}$.
3. Cell 3 interchanges one object 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 with cell 5 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.

3.2 Computational Efficiency

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. All the computation halt after $2n+3$ steps. Finally, the communication rules have length 5 at most. Therefore the main result of this paper holds.

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

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

These results hold from the previous construction 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 shed 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 [25] 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.

References

1. Alhazov, A., Cojocaru, S., Gheorghe, M., Rogozhin, Y., Rozenberg, G., Salomaa, A. (eds.): Membrane Computing - 14th International Conference, CMC 2013, Chişinău, Republic of Moldova, August 20-23, 2013, Revised Selected Papers, Lecture Notes in Computer Science, vol. 8340. Springer (2014)
2. Appel, K., Haken, W.: Every planar map is 4-colorable - 1: Discharging. Illinois Journal of Mathematics 21, 429–490 (1977)
3. Appel, K., Haken, W.: Every planar map is 4-colorable - 2: Reducibility. Illinois Journal of Mathematics 21, 491–567 (1977)
4. Christinal, H.A., Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: Tissue-like p systems without environment. In: Martínez del Amor, M.A., Păun, Gh., Pérez Hurtado, I., Riscos-Núñez, A. (eds.) Eighth Brainstorming Week on Membrane Computing. pp. 53–64. Fénix Editora, Sevilla, Spain (2010)
5. Cook, S.A.: The complexity of theorem-proving procedures. In: Proceedings of the Third Annual ACM Symposium on Theory of Computing. pp. 151–158. STOC '71, ACM, New York, NY, USA (1971)
6. Freund, R., Păun, Gh., Pérez-Jiménez, M.J.: Tissue P systems with channel states. Theoretical Computer Science 330(1), 101–116 (2005)

7. Garey, M.R., Johnson, D.S.: Computers and Intractability, A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, New York (1979)
8. Hartmanis, J.: Gödel, von Neumann and the P =? NP problem. In: Rozenberg, G., Salomaa, A. (eds.) Current Trends in Theoretical Computer Science - Essays and Tutorials, World Scientific Series in Computer Science, vol. 40, pp. 445–450. World Scientific (1993)
9. Jaffe, A.M.: The millennium grand challenge in mathematics. Notices of the American Mathematical Society 53(6), 652 – 660 (2006)
10. Krishna, S.N., Lakshmanan, K., Rama, R.: Tissue P systems with contextual and rewriting rules. In: Păun, Gh., Rozenberg, G., Salomaa, A., Zandron, C. (eds.) WMC-CdeA. Lecture Notes in Computer Science, vol. 2597, pp. 339–351. Springer, Berlin Heidelberg (2002)
11. Lakshmanan, K., Rama, R.: On the power of tissue P systems with insertion and deletion rules. In: Alhazov, A., Martín-Vide, C., Păun, Gh. (eds.) Preproceedings of the Workshop on Membrane Computing. pp. 304–318. Tarragona (July 17-22 2003)
12. Macías-Ramos, L.F., Pérez-Jiménez, M.J., Riscos-Núñez, A., Rius-Font, M., Valencia-Cabrera, L.: The efficiency of tissue P systems with cell separation relies on the environment. In: Csuhaj-Varjú, E., Gheorghe, M., Rozenberg, G., Salomaa, A., Vaszil, G. (eds.) Membrane Computing - 13th International Conference, CMC 2012, Budapest, Hungary, August 28-31, 2012, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7762, pp. 243–256. Springer (2012)
13. Martín-Vide, C., Pazos, J., Păun, Gh., Rodríguez-Patón, A.: A new class of symbolic abstract neural nets: Tissue P systems. In: Ibarra, O.H., Zhang, L. (eds.) COCOON. Lecture Notes in Computer Science, vol. 2387, pp. 290–299. Springer, Berlin Heidelberg (2002)
14. Martín-Vide, C., Păun, Gh., Pazos, J., Rodríguez-Patón, A.: Tissue P systems. Theoretical Computer Science 296(2), 295–326 (2003)
15. Mathu, T., Christinal, H.A., Díaz-Pernil, D.: A uniform family of tissue P systems with protein on cells solving 3-coloring in linear time. In: Calude, C.S., Dinneen, M.J. (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, vol. 9252, pp. 239–249. Springer (2015)
16. Pakash, V.: On the power of tissue P systems working in the maximal-one mode. In: Alhazov, A., Martín-Vide, C., Păun, Gh. (eds.) Preproceedings of the Workshop on Membrane Computing. pp. 356–364. Tarragona (July 17-22 2003)
17. Pan, L., Pérez-Jiménez, M.J.: Computational complexity of tissue-like P systems. Journal of Complexity 26(3), 296–315 (2010)
18. Pérez-Jiménez, M.J.: An approach to computational complexity in membrane computing. In: Mauri, G., Păun, Gh., Pérez-Jiménez, M.J., Rozenberg, G., Salomaa, A. (eds.) Workshop on Membrane Computing. Lecture Notes in Computer Science, vol. 3365, pp. 85–109. Springer (2004)
19. Pérez-Jiménez, M.J., Riscos-Núñez, A., Rius-Font, M., Romero-Campero, F.J.: A polynomial alternative to unbounded environment for tissue P systems with cell division. International Journal of Computer Mathematics 90(4), 760–775 (2013)
20. Pérez-Jiménez, M.J., Riscos-Núñez, A., Rius-Font, M., Valencia-Cabrera, L.: The relevance of the environment on the efficiency of tissue P systems. In: Alhazov et al. [1], pp. 308–321

21. Pérez-Jiménez, M.J., Riscos-Núñez, A., Romero-Jiménez, A., Woods, D.: Complexity - membrane division, membrane creation. In: Păun, Gh., Rozenberg, G., Salomaa, A. (eds.) The Oxford Handbook of Membrane Computing, pp. 302 – 336. Oxford University Press, Oxford, England (2010)
22. Păun, A., Popa, B.: P systems with proteins on membranes. Fundamenta Informaticae 72(4), 467–483 (2006)
23. Păun, A., Păun, M., Rodríguez-Patón, A., Sidoroff, M.: P systems with proteins on membranes: a survey. International Journal of Foundations of Computer Science 22(1), 39–53 (2011)
24. Păun, Gh., Pérez-Jiménez, M.J., Riscos-Núñez, A.: Tissue P systems with cell division. International Journal of Computers, Communication and Control 3(3), 295–303 (2008)
25. Song, B., Pan, L., Pérez-Jiménez, M.J.: Tissue P systems with protein on cells. Fundamenta Informaticae 144(1), 77–107 (2016)
26. Sosík, P.: Active membranes, proteins on membranes, tissue P systems: Complexity-related issues and challenges. In: Alhazov et al. [1], pp. 40–55
27. Sosík, P., Păun, A., Rodríguez-Patón, A.: P systems with proteins on membranes characterize PSPACE. Theoretical Computer Science 488, 78–95 (2013)
28. Stockmeyer, L.: Planar 3-colorability is NP-complete. SIGACT News 5(3), 19–25 (1973)
29. Zandron, C., Ferretti, C., Mauri, G.: Solving NP-complete problems using P systems with active membranes. In: Antoniou, I., Calude, C., Dinneen, M.J. (eds.) UMC. pp. 289–301. Springer (2000)