VisualTissue: A Friendly Tool to Study Tissue P Systems Solutions for Graph Problems

Rafael Borrego-Ropero, Daniel Díaz-Pernil, Juan A. Nepomuceno

Escuela Técnica Superior de Ingenería Informática
Universidad de Sevilla
Avda Reina Mercedes s/n, 41012 Sevilla, Spain
{rborrego,sbdani,janepo}@us.es

Summary. P systems can be classified in two main groups: P systems with the membrane structure described by a tree, and tissue P systems with the membranes placed in the nodes of an arbitrary graph. **NP**-complete problems have been solved in linear time by trading space for time in the framework of recognizing tissue P systems with cell division. The design of this kind of systems is not an easy task to understand. In this paper we present a software application to help the design of solutions to **NP**-complete problems in the framework of recognizing tissue P systems with cell division.

VisualTissue application can be downloaded from the web:
www.visualtissue.es.kz.

1 Introduction

Membrane Computing is an emergent branch of Natural Computing introduced by Gh. Păun in [9], which, considering as computations the processes that take place into living cells, constructs a new non-deterministic model of computation. In membrane computing there are two types of frameworks: P systems with the membrane structure described by a tree, inspired from the cell, and tissue P systems with the membranes placed in the nodes of an arbitrary graph. The second type corresponds to the idea of forming a network of membranes linked in a specific manner and working together, [10]. In both types, the main idea is having multisets of objects placed in compartments and evolving according to given rules in a synchronous non-deterministic maximally parallel manner.

In the last years, this new field has been addressed in different ways: the study of computational properties such as computational power or complexity classes, definition of new variants of membrane systems closer to biological reality, using Membrane Computing as a new framework for performing biological simulations, etc. In the P systems web page [18] several softwares can be found. Most of them are

thought in order to run an experiment which is built using some kind of membrane system as a simulation framework, for example, for simulating biological systems, as in [14]. On the other hand, another group of software applications are thought as an easy way of visualizing the computations of a P system, [8], that is to say, as a learning tool.

In this paper we present a new visual tool called *VisualTissue* which have been developed in order to help users to understand the computations of tissue P systems with cell division. We have considered the reference [4], which solves 3-col problem in the framework of tissue P systems with cell division. Polynomial solutions to **NP**-complete problems in Membrane Computing are done by trading time for space, in a theoretical way. Although real implementations of such systems are no possible, because it should be necessary to implement the maximal parallelism in some way, there are very interesting problems to threat in this framework, for example, the $\mathbf{P} \neq \mathbf{NP}$ conjecture [13]. Our tool helps the user to understand the performance of one class of such systems.

This paper is organized as follows: in Section 2 tissue P systems with cell division and the 3-col problem are explained, in Section 3 the *VisualTissue* application software is presented and some of the most important implementation aspects are commented. Finally, ideas for future work are formulated.

2 Tissue P Systems with Cell Division

We will give a definition of this model, but first we briefly recall some of the concepts used later.

2.1 Preliminaries

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

If $m = (A, f)$ is a finite multiset over A, then it will be denoted by $m = \{\{a_1, \ldots, a_k\}\}$, where each element a_i occurs $f(a_i)$ times, or by a string containing the symbols $a_1, \ldots, a_k$.

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

In what follows, we assume that the reader is already familiar with the basic notions and terminology underlying P systems. For details, see [10].

2.2 Tissue P Systems with Cell Division

In the first definition of the model of tissue P systems [6, 7] the membrane structure does not change along the computation. Based on the cell-like model of P systems with active membranes, Gh. Păun et al. presented in [12] 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 opposite holds in the cell-like model of P systems with active membranes, see [10]); 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:

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

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

The communication rules determine a virtual graph, where the nodes are the cells and the edges indicate if it is possible for pairs of cells to communicate directly. This is a dynamical graph, as new nodes can appear produced by the application of division rules.

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 arbitrarily large amount of copies.

A 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 to a cell i containing object a. The application of this rule divides this cell into two new cells with the same label. All the objects in the original cell are replicated and copied in each of the new cells, with the exception of the object a, which is replaced by the object b in the first new cell and by c in the second one.

Rules are used as usual in the framework of membrane computing, that is, in a maximally parallel way. In one step, each object in a cell 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.

2.3 Recognizing Tissue P Systems with Cell Division

NP-completeness has been usually studied in the framework of *decision problems*. Let us recall that a decision problem is a pair (I_X, θ_X) where I_X is a language over a finite alphabet (whose elements are called *instances*) and θ_X is a total boolean function over I_X.

In order to study the computing efficiency for solving **NP**-complete decision problems, a special class of tissue P systems with cell division is introduced in [12]: *recognizing tissue P systems*. The key idea of such recognizing system is the same one as from recognizing P systems with a cell-like structure.

Recognizing cell-like P systems were introduced in [15] 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, recognizing 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 cell*. 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 recognizing tissue P system with cell division of degree $q \geq 1$ is a tuple

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

where:

- $(\Gamma, w_1, \ldots, w_q, \mathcal{E}, \mathcal{R}, i_0)$ 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 some initial multisets $w_1, \ldots, w_q$, but not present in $\mathcal{E}$.
- Σ is an (input) alphabet strictly contained in Γ.
- $i_{in} \in \{1, \ldots, q\}$ is the input cell.
- The output region i_0 is the environment.
- All computations halt.
- If $\mathcal{C}$ is a computation of Π, then either the object `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}. 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. $\mathcal{C}$ is an accepting computation (respectively, rejecting computation) if the object `yes` (respectively, `no`) appears in the environment associated to the corresponding halting configuration of $\mathcal{C}$.

2.4 A Solution to the 3–coloring Problem

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

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

This problem is related to the famous Four Color Conjecture (solved by Appel and Haken [2, 3]). It is a particular case of the colorability problem: *Given an undirected graph G and a number k, decide whether G is k-coloreable.* The **NP**-completeness of the 3–coloring problem was proved by Stockmeyer [16] (see [5]).

First of all we define a polynomial encoding of the 3–coloring problem in a family $\mathbf{\Pi}$ of P systems constructed as in [4]. 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 $cod(u) = \{\{A_{ij} : \{A_i, A_j\} \in E \wedge 1 \leq i < j \leq n\}\} \cup \{\{w^q\}\}$.

The recognizing tissue P system with cell division that was used to solve the 3–coloring problem in [4] is defined as follows.

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

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

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

1. **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 $i = 1, \ldots, n$
2. **Communication rules:**
   $r_{3,i} \equiv (1, a_i/a_{i+1}, 0)$ for $i = 1, \ldots, 2n + m + \lceil log\ m \rceil + 10$
   $r_{4,i} \equiv (1, c_i/c_{i+1}^2, 0)$ for $i = 1, \ldots, 2n$
   $r_5 \equiv (1, c_{2n+1}/D, 2)$
   $r_6 \equiv (2, c_{2n+1}/d_1\overline{D}, 0)$
   $r_{7,i} \equiv (2, d_i/d_{i+1}^2, 0)$ for $i = 1, \ldots, \lceil log\ m \rceil$
   $r_8 \equiv (2, \overline{D}/e\, f_2, 0)$
   $r_{9,i} \equiv (2, f_i/f_{i+1}, 0)$ for $i = 2, \ldots, m + \lceil log\ m \rceil + 5$
   $r_{10,ij} \equiv (2, d_{\lceil log\ m \rceil + 1} A_{ij}/P_{ij}, 0)$ 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 $1 \leq j \leq n$
   $r_{19} \equiv (2, e\,\flat/\lambda, 0)$
   $r_{20} \equiv (2, e\ f_{m+\lceil log\ m \rceil + 6}/T, 0)$
   $r_{21} \equiv (2, T/\lambda, 1)$
   $r_{22} \equiv (1, b\,T/S, 0)$
   $r_{23} \equiv (1, S\ \texttt{yes}/\lambda, 0)$
   $r_{24} \equiv (1, b\ a_{2n+m+\lceil log\ m \rceil + 11}/N, 0)$
   $r_{25} \equiv (1, N\ \texttt{no}/\lambda, 0)$

- $\mathcal{E}(\langle n, m\rangle) = \Gamma(\langle n, m\rangle) - \{\texttt{yes}, \texttt{no}\}$
- $i_{in} = 2$ is the *input cell.*
- $i_0 = 0$ is the *output region.*

3 A Look Inside VisualTissue

VisualTissue is a visual software application to understand the design of solutions for **NP**-complete problems, in the framework of recognizing tissue P system with cell division.

Some programming decisions have been taken in order to develop the application. We have chosen *C#* as programming language because it is a portable and a powerful object-oriented programming language. A great graphical package is available with the language so that *C#* is a good language for developing a visual tool. The software follows the Model-View-Controller (MVC), an architecture model of software development used in interactive systems. Three different parts or layers can be distinguished: data handling layer, algorithmic or business logic layer, and user interface or graphical layer. With this, it is easier to do maintenance of the code.

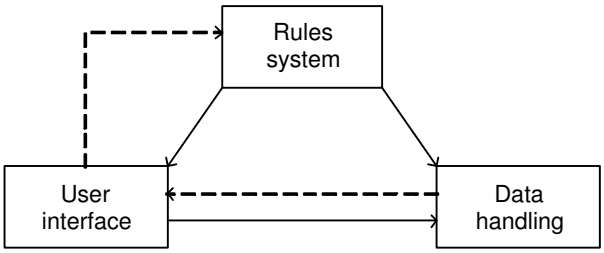

Fig. 1. Model-View-Controller architecture software

The application can handle data stored in XML and normal text files, which are compatible with *Algraf Project*, [19]. XML files can be easily generated and it is the best way to interact with data generated by other programs. Rules of the systems are fixed for the solution of each problem. In a future version, the user will be able to introduce his own tissue P system rules in order to study other possible solutions.

The algorithmic layer implements a recognizing tissue P system with cell division. All rules are applied in a non deterministic and maximal parallel way. The design of the tissue P system machine for solving the problem is non-deterministic until the $2n$ step. Every computation path reaches the same configuration in $2n$ steps, and after that the machine is deterministic and confluent. Consequently, we have chosen only a computation path in order to implement the tissue P system in the software, namely the one determined by the lexicographical election of the rules in non-deterministic steps. Other graph problems solutions can be easily added.

The graphical layer allows the user a visual and friendly interaction with the application. At the end of the simulation one can see the colored graph result of the problem if the answer is *yes*.

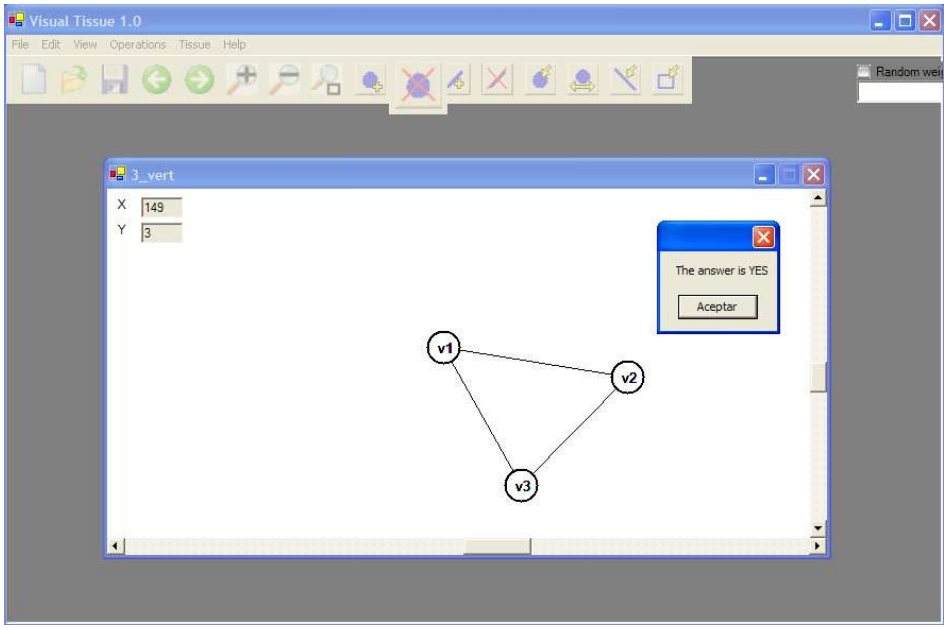

Fig. 2. Firstly, the graph to be studied must be drawn. A small screen show us wether it is 3-colorable or not.

3.1 An User Overview of the Application

This software tool allows us to follow step by step the execution of tissue P system with membrane division when solving the 3-col problem, [4]. The answers for a specific problem will be *yes* or *no*, and the colored graph is provided if it is possible. The user basically can do three different operations to run the system:

- Load or draw with the mouse the graph which is going to be studied. Several graphical options are available in the main window. Graphs can be handled in a easy way and images can be saved as image files or pdf files.
- Choose tissue 3-col algorithm in order to carry out the simulation. A second screen is showed in which each computation step can be observed. For each step, a graphical situation can be viewed and the different rules applied in each step are written in the bottom of the screen. The picture can be easily handled and each moment of the performance of the algorithm, and can be saved in different images format, or press the buttons with arrows to move between steps.
- Choose GO TO option to go directly to a specific step of the system.

VisualTissue software is available on the web, at: `www.visualtissue.es.kz`

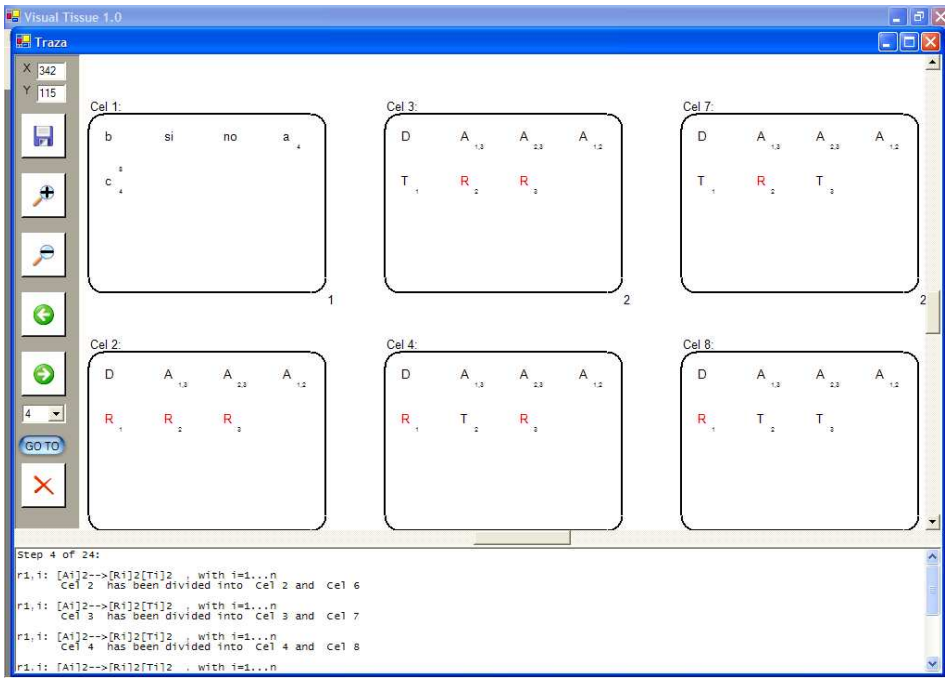

Fig. 3. The performance of the tissue P system can be observed step by step.

4 Future Works

Future works will be focused on different improvements as, for example: to handle input files of rules, work with other graph problems, consider other kinds of tissue P systems rules such as membrane creation, etc. One of the most interesting future tasks will be to build a software which simulates a Spiking Neural P system, extending this graphical tool. This future software task will be not only a way of visualizing the performance of the system, but also a framework to do simulations with Spiking Neural P systems too.

References

1. B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, P. Walter: *The Molecular Biology of the Cell*. Fourth Edition, Garland Publ. Inc., London, 2002.
2. K. Appel, W. Haken: Every planar map is 4-colorable - 1: Discharging. *Illinois Journal of Mathematics*, 21 (1977), 429–490.
3. K. Appel, W. Haken: Every planar map is 4-colorable - 2: Reducibility. *Illinois Journal of Mathematics*, 21 (1977), 491–567.
4. D. Díaz-Pernil, M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Núñez: A linear–time tissue P system based solution for the 3–coloring problem. *Theoretical Computer Science*, to appear.

5. M.R. Garey, D.S. Johnson: *Computers and Intractability A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.
6. C. Martín Vide, J. Pazos, Gh. Păun, A. Rodríguez Patón: A new class of symbolic abstract neural nets: Tissue P systems. *Lecture Notes in Computer Science* **2387** (2002), 290–299.
7. C. Martín Vide, J. Pazos, Gh. Păun, A. Rodríguez Patón: Tissue P systems. *Theoretical Computer Science*, 296 (2003), 295–326.
8. I.A. Nepomuceno-Chamorro: A Java simulator for membrane computing. *Journal of Universal Computer Science*, 10 (2001), 620–629.
9. Gh. Păun: Computing with membranes. *Journal of Computer and System Sciences*, 61, 1 (2000), 108–143.
10. Gh. Păun: *Membrane Computing. An Introduction*. Springer-Verlag, Berlin, 2002.
11. Gh. Păun, M.J. Pérez-Jiménez: Recent computing models inspired from biology: DNA and membrane computing. *Theoria*, 18 (2003), 72–84.
12. Gh. Păun, M.J. Pérez-Jiménez, A. Riscos-Núñez: Tissue P system with cell division. In Gh. Păun, A. Riscos-Núñez, A. Romero-Jiménez and F. Sancho-Caparrini (eds.), *Second Brainstorming Week on Membrane Computing*, Sevilla, Report RGNC 01/2004, 2004, 380–386.
13. 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.) *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 3365 (2005), 85–109.
14. M.J. Pérez-Jiménez, F.J. Romero-Campero: P systems, a new computational modelling tool for systems biology. *Transactions on Computational Systems Biology VI. Lecture Notes in Bioinformatics*, Springer-Verlag, Berlin, 4220 (2006), 176-197
15. M.J. Pérez-Jiménez, A., Romero-Jiménez, F. Sancho-Caparrini: A polynomial complexity class in P systems using membrane division. In E. Csuhaj-Varjú, C. Kintala, D. Wotschke and Gy. Vaszil (eds.), *Proceedings of the 5th Workshop on Descriptional Complexity of Formal Systems, DCFS 2003*, 2003, 284–294.
16. L.J. Stockmeyer: Planar 3-colorability is NP-complete. *SIGACT News*, 5, 3 (1973), 19–25.
17. ISI web page `http://esi-topics.com/erf/october2003.html`
18. P systems web page `http://psystems.disco.unimib.it/`
19. Algraf Project web page `http://www.algraf.es.kz/`