A Cellular Way to Obtain Homology Groups in Binary 2D Images

Daniel Díaz-Pernil[1], Miguel A. Gutiérrez-Naranjo[2],
Pedro Real[1], Vanesa Sánchez-Canales[1]

[1] Research Group on Computational Topology and Applied Mathematics
Department of Applied Mathematics
University of Sevilla
{sbdani,real,vscanales}@us.es
[2] Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla
magutier@us.es

Summary. In this paper we present a P systems-based solution for the *Homology Groups of Binary 2D Image (HGB2I) Problem*, a classical problem in Homology Theory. To this aim, we present a family of P systems which solves all the instances of the problem in the framework of *Tissue-like P systems with catalysts*. This new framework combines the membrane structure and symport-antiport communication rules of tissue-like P systems with the power of catalysts and inhibitors.

1 Introduction

Homology theory is a branch of Algebraic Topology that attempts to distinguish between spaces by constructing algebraic invariants that reflect the connectivity properties of the space. The field has its origins in the work of the French mathematician, theoretical physicist, and a philosopher of science Jules Henri Poincaré. Homology groups (related to the different n-dimensional holes, connected components, tunnels, cavities, etc., of a geometric object) are invariants from Algebraic Topology which are frequently used in Digital Image Analysis and Structural Pattern Recognition. In some sense, they reflects the topological nature of the object in terms of the number and characteristics of its holes.

In this paper we explore one of the main problems from Homology Theory in terms of Membrane Computing[3]. The chosen problem is the *Homology Groups of Binary 2D Image (HGB2I) Problem*: Given a binary 2D digital image, calculate the number of black connected components and the representative curves of the

[3] We refer to [20] for basic information in this area, to [23] for a comprehensive presentation and the web site [17] for the up-to-date information.

holes of these components. We can divide this problem in two sub-problems, H_0 problem (number of black connected components) and H_1 problem (number of holes). This problem and the way of computing and representing topological information (neighborhood, connectedness, orientation, etc.) form an important part in applications such as image classification, indexing, shape description and shape recognition.

This is not the first bio-inspired approach to problems to Algebraic Topology. In 1996, J. Chao and J. Nakayama connected Natural Computing and Algebraic Topology using Neural Networks [5] by extended Kohonen mapping. Some years after, Subramanian *et al.* presented in [3, 4] two works where Digital Image and Natural Computing were linked. Our paper can be seen as a new step from the work by Cristinal *et al.* [6, 7] in their effort for bridging Membrane Computing and Algebraic Topology.

The solution presented in this paper to the HGB2I problem has been designed in a new P system framework called *tissue-like P systems with catalysts.* It takes the membrane structure and symport-antiport communication rules with the power of catalysts and inhibitors.

Time to calculate the homology groups of 2D digital images with these P systems is logarithmic with respect to the input data with size n^2. This involves an improvement with regard to the algorithms development by S. Peltier et al. in [24], where they use irregular graphs pyramids with a time complexity of $\mathrm{O}(\mathrm{n}^{5/3})$.

The paper is organized as follows: In the next section we formally present the framework of *tissue-like P systems with catalysts.* In Section 3, we show how these P systems can be used to solve the H_0 and H_1 problems in Homology Theory. Next we will show a pair of examples. The paper ends with some final remarks and open lines for the future.

2 Tissue-like P Systems with Catalysts

Tissue P systems were presented in [15, 16]. This P system model is inspired in the intercellular communication and cooperation between neurons. The mathematical model of these devices 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[4]. Symport rules move objects across a membrane together in one direction, whereas antiport rules move objects across a membrane in opposite directions.

In tissue-like P systems the membrane structure is a general undirected graph. The edges of such graph are not given explicitly, but they are deduced from the set of rules. From the seminal definition of tissue P systems, several research lines have been developed and other variants have arisen (see, for example, [1, 2, 10, 12, 14, 18, 22]).

[4] This way of communication for P systems was introduced in [21].

Catalytic P systems were introduced in [19]. The main feature of these P systems is the presence of objects in membranes such that they are not consumed by the application of the rule, but their presence in the membrane is necessary for the triggering. Catalysts have been deeply studied in Membrane Computing (see, e.g. [8, 13, 11]), but to the best of our knowledge, this is the first time in which catalysts are used for tissue P systems[5].

Next we provide the definition of tissue-like P systems with catalysts:

Definition 1. *A tissue-like P system with catalyst of degree* $q \geq 1$ *is a tuple of the form*

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

where:

1. Γ *is a finite alphabet, whose symbols will be called objects.*
2. $\mathcal{E} \subseteq \Gamma$ *is a finite alphabet representing the set of the objects in the environment available in an arbitrary large amount of copies.*
3. $w_1, \dots, w_q$ *are strings over* Γ *representing the multisets of objects associated with the cells in the initial configuration.*
4. $\mathcal{R}$ *is a finite set of enzymatic rules of the following form:* $(\neg in, cat \,|\, i, u/v, j)$ *for* $i, j \in \{0, 1, 2, \dots, q\}, i \neq j$*,* $in, cat,\ u, v \in \Gamma^*$*. The length of a communication rule is defined as* $|u| + |v|$*. The catalyst and the inhibitor are not modified by the application of the rules and* cat*,* in *and* v *can be empty.*
5. $i_0 \in \{0, 1, 2, \dots, q\}$ *denotes the output region, which can be the environment* ($i_0 = 0$) *or the region inside a cell* ($1 \leq i_0 \leq q$)*.*

Informally, a tissue-like P system with catalysts of degree $q \geq 1$ can be seen as a set of q cells (each one consisting of a single membrane) labeled by $1, 2, \dots, q$. The cells are the nodes of a virtual graph, where the edges connecting the cells are determined by the communication rules of the system (an edge linking two cells indicates that they are able to trade objects between them). In our definition, all objects in the alphabet can act as catalyst or inhibitor, depending on the applied rule. This means that the inhibitor or catalysts for a rule can be sent to another cell (or to the environment) by another rule.

The enzymatic rule $(\neg in, cat \,|\, i, u/v, j)$ can be applied over two cells (or a cell and the environment) i and j such that u (contained in cell i) is traded against v (contained in cell j). The rule is applied if in membrane with label i the objects of the set *cat* are present (catalyst) and none of the objects from the set *in* are present (inhibitors). If the catalyst and the inhibitor are empty, then the rule is called a *communication rule*.

Rules are used as usual in the framework of membrane computing, that is, in a maximally parallel way (a universal clock is considered). 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 multiset of rules.

[5] Comprehensive information about catalytic P systems can be found at [9].

A *configuration* is an instantaneous description of the system Π, and it is represented as a tuple $(w_0, w_1, \ldots, w_q)$. Given a configuration, we can perform a computation step and obtain a new configuration by applying the rules in a parallel manner as it is shown above. A sequence of computation steps is called a *computation.* A configuration is *halting* when no rules can be applied to it. The output of a computation is collected from its halting configuration by reading the objects contained in the output cell.

3 Using Tissue-like P Systems to Obtain H_0 and H_1

As pointed out above, given a binary 2D digital image, the problem consists on calculating the number of black connected components and the representative curves of the holes of these components. We can divide this problem in two subproblems, the H_0 *problem* consists on calculating the number of black connected components and the H_1 *problem*, which consists on calculating the number of holes. In this paper we present a new technique to use tissue-like P systems with catalyst to obtain homological information of binary 2D digital images.

A 2D digital image I can be considered like a matrix where one pixel is an element of the matrix. If we have pixels the following question is to know when two pixels are *adjacent* (connected). There exists two natural possibilities, consider to work with a 4-adjacency (Von Neumann neighborhood in cellular automata) or 8-adjacency (Moore neighborhood in cellular automata).

In the first case, given a pixel K_{ij} (where $K = B \vee K = W$), the list of adjacent pixels to this is $\{K_{ij-1}, K_{ij+1}, K_{i-1j}, K_{i+1j}\}$ i.e.; the adjacent pixels to any pixel $K_{i,j}$ are just north, south, west, east of this (no in the diagonal respect to considered pixel), such we can observe in the following:

$$\begin{array}{ccc} & B & \\ B & K & W \\ & W & \end{array}$$

In the second we consider the pixel K_{ij} (where $K = B \vee K = W$), the list of adjacent pixels to this is $\{K_{i-1j-1}, K_{i-1j}, K_{i-1j+1}, K_{ij-1}, K_{ij+1}, K_{i+1j-1}, K_{i+1j}, K_{i+1j+1}\}$ i.e.; the adjacent pixels to a any pixel $K_{i,j}$ are just up, down, right and left of this and, moreover, we consider the diagonal objects, such we can observe in the following:

$$\begin{array}{ccc} B & W & B \\ W & K & B \\ B & B & W \end{array}$$

We have decide to consider in this paper the 4-adjacency for black pixels and the 8-adjacency for white pixels.

Given an image I with size n^2, we divide I in a set of pixels, black or white, but not both. We codify each pixel (i, j) by the object a_{ij} where $a = b$ (black) or $a = w$ (white). We can assign to each object a label associated to the codified pixel by this object. So, we have the objects of the form $(a_{ij}, (i, j))$. We will see below how to use these labels to solve our problem with P systems.

3.1 Solving H_0 Problem

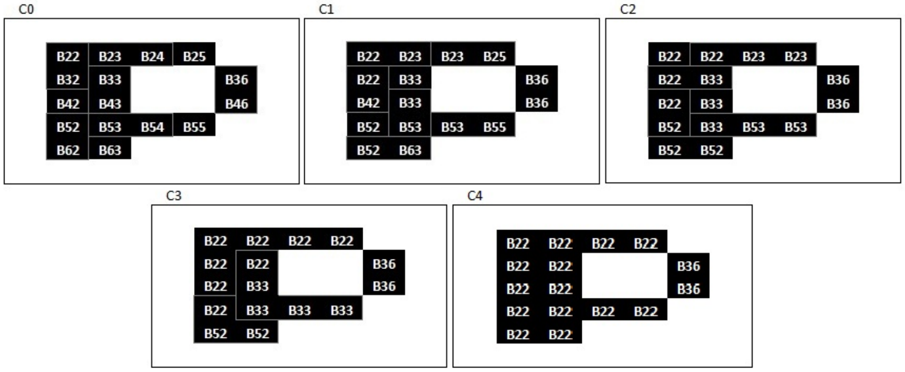


Fig. 1. A simple example to obtain H_0

In order to provide a logarithmic-time uniform solution to the H_0 problem, we design a family of tissue-like P systems with catalyst, Π_0. Given an image I of size n^2, we take the system of the family $\Pi_0(n)$ to work with I. The input data (image I) is codified by a set of objects b_{ij} and w_{ij} for $1 \le i, j \le n$. Each pixel of the image is given by an object z_{ij} with $z = b$ or $z = w$.

The family of P systems is defined as follows:

$$\Pi_0(n) = (\Gamma, \Sigma, \mathcal{E}, \omega_1, \omega_2, \mathcal{R}, i_{in}, i_0)$$

where:

- $\Gamma = \{a_i : 1 \le i \le n+2\} \cup$
  $\{b_{ij}, w_{ij} : 1 \le i, j \le n\} \cup$
  $\{(b_{ij}, (k, l)) : (1,1) \le (i, j) \le (k, l) \le (n, n)\} \cup$
  $\{A_{ijkl} : (1,1) \le (i, j) < (k, l) \le (n, n)\}$.
- $\Sigma = \{b_{ij}, w_{ij} : 1 \le i, j \le n\}$.
- $\omega_1 = \{a_1\}$.
- $\omega_2 = \emptyset$.
- $\mathcal{E} = \Gamma - \Sigma$.
- $\mathcal{R}$ is the set of rules:
  - ○ $R_1 \equiv (1, a_i/a_{i+1}, 0)$ for $1 \le i \le n+1$.
    These rules generate a counter that will be used in the output of the system.

  - ○ $R_2 \equiv (1, b_{ij}/(b_{ij},(i,j)), 0)$ for $1 \leq i,j \leq n$.
    These rules add labels to black pixels in order to work with them.
  - ○ $R_3 \equiv (1, (b_{ij},(k,l))(b_{i'j'},(k',l'))/(b_{ij},(k,l))(b_{i'j'},(k,l))A_{klk'l'}, 0)$ for $(1,1) \leq (k,l) < (k',l') \leq (n,n)$, and (i,j), (i',j') adjacent pixels.
  - ○ $R_4 \equiv (1, (b_{ij},(k,l))(b_{i'j'},(k',l'))/(b_{ij},(k',l'))(b_{i'j'},(k',l'))A_{k'l'kl}, 0)$ for $(1,1) \leq (k',l') < (k,l) \leq (n,n)$ and (i,j), (i',j') adjacent pixels.
    The two last types of rules change the labels of adjacent pixels, we need all the adjacent black pixels to have the same label, so we will know that they are all in the same connected component.
  - ○ $R_5 \equiv (A_{ijkl}|1, (b_{i'j'},(k,l))/(b_{i'j'},(i,j)), 0)$ for $1 \leq i,j,k,l,i',j' \leq n$.
    In these rules we introduce catalysts, and process becomes faster. The catalyst has been created when the pixel labeled by (k,l) traded its label for (i,j), so (i,j) and (k,l) are adjacent pixels and other pixels with these labels can be changed.
  - ○ $R_6 \equiv (a_{n+2}|1, (b_{ij},(i,j))/\lambda, 2)$.
    With these rules we send one pixel for each connected component to the cell 2.
- $i_{in} = 1$ is the input cell.
- $i_0 = 2$ is the output cell.

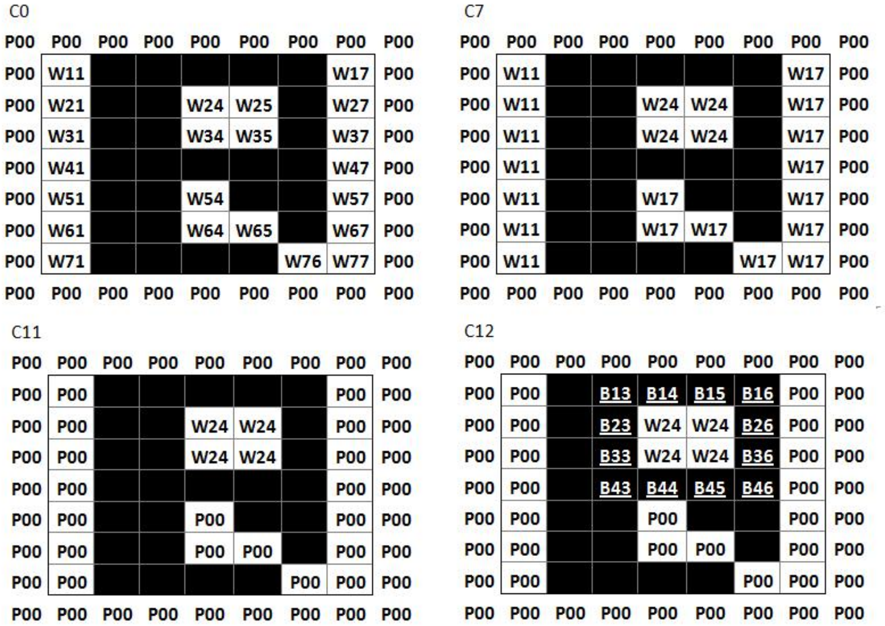
Each system of the family implements the following stages:

1. *Label Allocation Stage*: Cell 1 trades objects b_{ij} against others with the form $(b_{ij},(i,j))$ with the environment. The white objects are not transformed.
2. *Label Conversion Stage*: We can compare the black adjacent pixels by using catalyst, and we trade the label of the greatest pixel against the label of the other pixel; i.e. $(i,(b_{ij},(i',j'))(b_{kl},(k',l'))/(b_{ij},(i',j'))(b_{kl},(i',j'))A_{i'j'k'l'},j)$, where (i,j) and (k,l) are adjacent pixels. Moreover, we can see a new object arriving to cell i. It is a catalyst and it is used to codify if two labels must be compared. Later, they are connected, and one of them can be changed by the other one, as we can see in the Figure 1.
3. *Answer Stage*: In the step $n+2$, the object a_{n+2} arrives to the cell 1 due to the counter. It is used by the system as a catalyst, and the objects with the form $(b_{ij},(i,j))$ are sent to the output cell representing each one to a black connected component. The P system have used $n+2$ steps to obtain the number of black connected components of an n^2 image.

Figure 1 shows a computation of the $\Pi_0(7)$ system whose input data is the configuration $C0$ of the picture.

3.2 Solving H_1 Problem

With respect to the H_1 problem we use the same technique that we present above where labels are associated to the objects codifying pixels. We can construct a



C0

P00	P00	P00	P00	P00	P00	P00	P00	P00
P00	W11	■	■	■	■	■	W17	P00
P00	W21	■	■	W24	W25	■	W27	P00
P00	W31	■	■	W34	W35	■	W37	P00
P00	W41	■	■	■	■	■	W47	P00
P00	W51	■	■	W54	■	■	W57	P00
P00	W61	■	■	W64	W65	■	W67	P00
P00	W71	■	■	■	■	W76	W77	P00
P00	P00	P00	P00	P00	P00	P00	P00	P00

C7

P00	P00	P00	P00	P00	P00	P00	P00	P00
P00	W11	■	■	■	■	■	W17	P00
P00	W11	■	■	W24	W24	■	W17	P00
P00	W11	■	■	W24	W24	■	W17	P00
P00	W11	■	■	■	■	■	W17	P00
P00	W11	■	■	W17	■	■	W17	P00
P00	W11	■	■	W17	W17	■	W17	P00
P00	W11	■	■	■	■	W17	W17	P00
P00	P00	P00	P00	P00	P00	P00	P00	P00

C11

P00	P00	P00	P00	P00	P00	P00	P00	P00
P00	P00	■	■	■	■	■	P00	P00
P00	P00	■	■	W24	W24	■	P00	P00
P00	P00	■	■	W24	W24	■	P00	P00
P00	P00	■	■	■	■	■	P00	P00
P00	P00	■	■	P00	■	■	P00	P00
P00	P00	■	■	P00	P00	■	P00	P00
P00	P00	■	■	■	■	P00	P00	P00
P00	P00	P00	P00	P00	P00	P00	P00	P00

C12

P00	P00	P00	P00	P00	P00	P00	P00	P00
P00	P00	■	B13	B14	B15	B16	P00	P00
P00	P00	■	B23	W24	W24	B26	P00	P00
P00	P00	■	B33	W24	W24	B36	P00	P00
P00	P00	■	B43	B44	B45	B46	P00	P00
P00	P00	■	■	P00	■	■	P00	P00
P00	P00	■	■	P00	P00	■	P00	P00
P00	P00	■	■	■	■	P00	P00	P00
P00	P00	P00	P00	P00	P00	P00	P00	P00

Fig. 2. Representative configurations of a simple example to obtain H_1

family of tissue-like P systems with catalyst, Π_1, to obtain a solution of the H_1 problem. Moreover, we can obtain the curves formed by black pixels containing the holes of the black connected components of the input image. So, we introduce in this paper a technique for segmenting images using catalysts.

Given an image I of size n^2 we take the system of the family $\Pi_1(n)$ to work with I. The input data (image I) is codified by a set of following objects: b_{ij} and w_{ij} for $1 \leq i, j \leq n$. Then, each pixel of the image is given by an object z_{ij} with $z = b \vee w$. The family of P systems is defined as follows:

$$\Pi_1(n) = (\Gamma, \Sigma, \mathcal{E}, \omega_1, \omega_2, \mathcal{R}_1, \ldots, \mathcal{R}_{10}, \{\mathcal{R}_6, \mathcal{R}_8\} > \mathcal{R}_1, i_{in}, i_0)$$

where:

- $\Gamma = \{z_i : 1 \leq i \leq n+3\} \cup$
  $\{b_{ij}, \bar{b}_{ij}, w_{ij}, (w_{ij}, (k,l)) : 1 \leq i, j, k, l \leq n\} \cup$
  $\{(p_{ij}, (0,0)), (p_{ji}, (0,0)) : i = 0, n+1, 0 \leq j \leq n+1\} \cup$
  $\{Z_{ijkl} : (1,1) \leq (i,j) < (k,l) \leq (n,n)\}$.
- $\Sigma = \{b_{ij}, w_{ij} : 1 \leq i, j \leq n\}$.
- $\mathcal{E} = \Gamma - \Sigma$.
- $\omega_1 = \{z_1, (p_{ij}, (0,0)), (p_{ji}, (0,0)) : i = 0, n+1, 0 \leq j \leq n+1\}$.
- $\omega_2 = \emptyset$.

- The sets of rules are:
  - $R_1 \equiv (1, z_i/z_{i+1}, 0)$ for $1 \leq i \leq n+5$.

    This rule counts the number of steps of the process. We will use this to start the *Deleting Stage* after $n+2$ steps, and the *Segmenting Stage* after $n+4$ steps.
  - $R_2 \equiv (1, w_{ij}/(w_{ij}, (i,j)), 0)$ for $1 \leq i, j \leq n$.

    These are the only rules used in the *Label Allocation Stage*. These rules add labels to white pixels in order to work with them.
  - $R_3 \equiv (1, (w_{ij}, (k,l))(w_{i'j'}, (k',l'))/(w_{ij}, (k,l))(w_{i'j'}, (k,l))Z_{klk'l'}, 0)$ for $(1,1) \leq (k,l) < (k',l') \leq (n,n)$, $w_{ij}, w_{i'j'}$ adjacent pixels.
  - $R_4 \equiv (1, (w_{ij}, (k,l))(w_{i'j'}, (k',l'))/(w_{ij}, (k',l'))(w_{i'j'}, (k',l'))Z_{k'l'kl}, 0)$ for $(1,1) \leq (k',l') < (k,l) \leq (n,n)$, $b_{ij}, b_{i'j'}$ adjacent.

    These two set of rules are used in *Label Conversion Stage* to compare two adjacent white pixels, and change the label of one of them. We need all the adjacent white pixels to have the same label.
  - $R_5 \equiv (Z_{ijkl}|1, (w_{i'j'}, (k,l))/(w_{i'j'}, (i,j)), 0)$ for $1 \leq i, j, k, l, i', j' \leq n$.

    The catalyst Z_{ijkl} acts to become the process faster. It has been created when the pixel labeled by (k,l) traded its label for (i,j), so (i,j) and (k,l) are adjacent pixels and other pixels with these labels can be changed.
  - $R_6 \equiv (z_{n+3}|1, (p_{ij}, (0,0))(w_{kl}, (k',l'))/(p_{ij}, (0,0))(p_{kl}, (0,0))Z_{00kl}, 0)$ for $(i,j), (k,l)$ 8-adjacent pixels, $0 \leq i, j \leq n+1$, $1 \leq k, l, k', l' \leq n$.

    These rules are used in *Deleting Stage* to delete white pixels which are out of the connected black component. By using 8-adjacency, we become outer white pixels into pink pixels, in order to differentiate them from the interior white pixels (holes). We will refer to the objects p_{ij} as *pink* pixels.
  - $R_7 \equiv ((Z_{00ij}|1, (w_{i'j'}, (i,j))/(p_{i'j'}, (0,0)), 0)$.

    A new catalyst acts in the same way, trading white exterior pixel for pink pixels. In this way, the *Deleting Stage* takes only 2 step.
  - $R_8 \equiv (z_{n+5}|1, (w_{ij}, (i',j'))b_{kl}/(w_{ij}, (i',j'))\bar{b}_{kl}, 0)$ for w_{ij}, b_{kl} 8-adjacent pixels $1 \leq i'.j', i, j, k, l \leq n$.

    In the *Segmenting Stage* a black pixel is marked if it and a white pixel are 8-adjacent pixels. It starts after $n+2$ steps.
  - $R_9 \equiv (1, \bar{b}_{ij}/\lambda, 2)$ for $1 \leq i, j \leq n$.

    At the end, in the
  - *Answer Stage*, black marked pixels are sent to membrane number 2, so we obtain which black pixels are containing the holes.
  - $R_{10} \equiv (z_{n+6}|1, (w_{ij}, (i,j))/\lambda, 2)$ for $1 \leq i, j \leq n$.

    We want to obtain the number of holes too, so these rules send one white pixel for each hole to membrane number 2.
- $i_{in} = 1$ is the input cell.
- $i_0 = 2$ is the output cell.

We will also use priorities among rules. Rules from sets $\mathcal{R}_6$ and $\mathcal{R}_8$ are applied before rules from the set $\mathcal{R}_1$.

The computation of each P system of the family has the following phases:

1. *Label Allocation Stage*: Cell 1 trades objects w_{ij} against others with the form $(w_{ij},(i,j))$ with the environment.
2. *Label Conversion Stage*: We compare the label of two white adjacent pixels, and we trade the label of the greatest pixel against the label of the other pixel; i.e., we use rules with the form $(i,(w_{ij},(i',j'))(w_{kl},(k',l'))/(w_{ij},(i',j'))(w_{kl},(i',j'))Z_{i'j'k'l'},j)$, where (i,j) and (k,l) are adjacent pixels. Moreover, we can see a new object arriving to cell i, $Z_{i'j'k'l'}$. It is a catalyst and is used to codify when two labels must be compared. Then, the labels are connected, and one of them can be changed by the other one, as we can see in $C7$ in the Figure 2.
3. *Deleting Stage*: Initially, system keeps in cell 1 a set of objects codifying the frame of the input image ($p_{0i}, p_{n+1i}, p_{i0}, p_{in+1}$ for $i = 0,\ldots,n+1$) with the label $(0,0)$ associated. When the input data is introduced in the system, the white pixels not contained inside of black connected components are sent to the environment to trade against of objects with the form of the frame. We need a linear number of steps with respect to n to eliminate all the possible white pixels. We can see the result in $C11$ in the Figure 2.
4. *Segmenting Stage*: This part begins when deleting stage finishes due to the counter z_i (rules R_1). If there are white pixels in cell 1 in this step are in a hole. The P system takes pairs of adjacent pixels, one black and the other white, adding a mark to the black pixels of these pairs. Then, we have marked the black pixels adjacent to a hole. We need a constant number of steps to segment an image with P systems. Figure 2 shows in $C12$ how the holes of the image are codified.
5. *Answer Stage*: We send the marked black pixels to output cell in the following step to be marked. So, we obtain, the representative curves of the holes in the image I. We also send white pixels which keep their labels, there is only one pixel for each connected white component, ie, for each hole in the image. We only need one step more with respect to the segmenting stage.

3.3 Complexity and Necessary Resources

Bearing in mind the size of the input data is $O(n^2)$, the amount of necessary resources for defining the systems of our two families and the complexity of our problems can be observed in the following table:

HGB2I Problem

	$\mathbf{H_0}$ Problem	$\mathbf{H_1}$ Problem
Complexity		
Number of steps of a computation	$n+2$	$n+7$
Necessary Resources		
Size of the alphabet	$O(n^4)$	$O(n^4)$
Initial number of cells	2	2
Initial number of objects	1	$O(n)$
Number of rules	$O(n^6)$	$O(n^6)$
Upper bound for the length of rules of the systems	5	5

4 Final Remarks

Problems associated with the treatment of Digital Images have several interesting features from the Membrane Computing point of view. One of them is that they can be suitable for parallel processing. In many cases, the same sequential algorithm must be applied in different regions of the image which are independent. Other important feature is that the information of the image can be split into little pieces of information and the local transformations can be processed by re-writing-type rules.

These features lead us to explore the possibilities of using Membrane Computing techniques to well-known problems in Digital Images. In this paper we provide a solution in the framework of tissue P systems with catalysts, but a deeper study is necessary. The research lines related to the most suitable P system model for Homology Theory problems or which are the most relevant features of P systems which can represent the nature of the problems are open.

Acknowledgement

The first and second authors acknowledge the support of the projects TIN2008-04487-E and TIN-2009-13192 of the Ministerio de Ciencia e Innovación of Spain and the support of the Project of Excellence with *Investigador de Reconocida Valía* of the Junta de Andalucía, grant P08-TIC-04200. The third and fourth authors acknowledge the support of the project MTM2006-03722 of the Ministerio español de Educación y Ciencia and the project PO6-TIC-02268 of Excellence of Junta de Andalucía.

References

1. 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.

2. 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.
3. Ceterchi, R., Madhu, M., Paun, G., Subramanian, K.G. Array-rewriting P systems. *Natural Computing*, **2**, (2003), 229–249.
4. Chandra, P.H., Subramanian, K.G. On Picture Arrays Generated by P Systems, *Preproceedings of WMC6*, (2005), 282–288.
5. Chao. J. and Nakayama. J. Cubical Singular Simples Model for 3D Objects and Fast Computation of Homology Groups. *Proceedings of ICPR'96 IEEE*, (1996), 190–194 .
6. Christinal, H.A., Díaz-Pernil, D., Real, P. Segmentation in 2D and 3D Image Using Tissue-Like P System. *Lecture Notes in Computer Science*, **5856**, (2009) 169–176.
7. Christinal, H.A., Díaz-Pernil, D., Real, P. Using Membrane Computing for Obtaining Homology Groups of Binary 2D Digital Images. *Lecture Notes in Computer Science*, **5852**, (2009), 388–401.
8. Freund, R., Kari, L. Oswald, M., Sosik, P.Computationally universal P systems without priorities: two catalysts are sufficient, *Theoretical Computer Science*, **330**, (2005), 251–266.
9. Freund, R., Ibarra, O., Păun, A., Sosík, P., Yen, H.-C. Catalytic P Systems. In [23], 83–117.
10. Freund, R., Păun, Gh. and Pérez-Jiménez, M.J. Tissue P Systems with channel states. *Theoretical Computer Science*, **330**, (2005), 101–116.
11. Krishna, S.N. On Pure Catalytic P Systems. *Lecture Notes in Computer Science* **4135**, (2006), 152–165.
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. Krishna, S.N., Păun, A. Results on catalytic and evolution-communication P systems. *New Generation Computing*, **22**, 4 (2004), 377–394.
14. Lakshmanan K. and Rama, R. On the Power of Tissue P Systems with Insertion and Deletion Rules. *Preproceedings of WMC*, (2003), 304–318.
15. 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.
16. Martín Vide, C. Pazos, J. Păun, Gh. and Rodríguez Patón, A. Tissue P systems. *Theoretical Computer Science*, **296**, (2003), 295–326.
17. P systems web page `http://ppage.psystems.eu/`
18. Prakash, V.J. On the Power of Tissue P Systems Working in the Maximal-One Mode. *Preproceedings of WMC*, (2003), 356–364.
19. Păun, Gh. Computing with Membranes. *Journal of Computer and System Sciences.* **61**, (2000), 108–143.
20. Păun, Gh. *Membrane Computing. An Introduction.* Springer–Verlag, Berlin, (2002).
21. Păun, A. and Păun, Gh. The power of communication: P systems with symport/antiport. *New Generation Computing*, **20**, 3, (2002), 295–305.
22. Păun, Gh., Pérez-Jiménez, M.J. and Riscos-Núñez, A. Tissue P System with cell division. *BWMC2*, (2004), 380–386.
23. Păun, Gh. Rozenberg, G. Salomaa, A. eds: Handbook of Membrane Computing. *Oxford University Press*, (2009).
24. Peltier, S., Ion, A., Haxhimusa, Y., Kropatsch, W.G. and G. Damiand.: Computing Homology Group Generators of Images Using Irregular Graph Pyramids. *Lecture Notes in Computer Science*, **4538**, (2007), 283–294.