ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **13**, Number 2, 2010, 141–152

Computing Homology Groups in Binary 2D Imagery by Tissue-like P Systems

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, Spain

E-mail: {sbdani,real,vscanales}@us.es

[2]Research Group on Natural Computing,
Department of Computer Science and Artificial Intelligence,
University of Sevilla, Spain

E-mail: magutier@us.es

Abstract. We present a new solution for the *Homology Groups of Binary 2D Image (HGB2I) Problem* by using Membrane Computing techniques. This is a classical problem in *Homology Theory* which tries to calculate the number of connected components and the representative curves of the holes of these components from a given binary 2D image. 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 is a new framework which combines the membrane structure and symport-antiport communication rules of tissue-like P systems with the power of catalysts.

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 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 reflect 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[1]. The chosen problem is the *Homology Groups of Binary 2D Image (HGB2I) Problem*: Given a binary 2D digital image, calculate the number of connected components and the representative curves of the holes of these components. The algorithms for solving this problem have an immediate application in Digital Imagery, since the topological information is one of the key points in image classification, indexing, shape description and shape recognition.

This is not the first bio-inspired approach to problems in Algebraic Topology. In 1996, Chao and 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 combines the membrane structure and symport-antiport communication rules with the power of catalysts.

The 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 developed by Peltier *et al.* in [21], where they use irregular graphs pyramids with a time complexity of $O(n^{5/3})$.

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 and illustrate the solution with 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 introduced in [14, 15]. This P system model is inspired by 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 [16]. 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

[1] We refer to [18] for basic information in this area, to [20] for a comprehensive presentation and the web site [22] for the up-to-date information.

been developed and other variants have arisen (see, for example, [1, 2, 10, 12, 19]).

Catalytic P systems were introduced by Păun in [17]. 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 triggering the rules. Catalysts have been deeply studied in Membrane Computing (see, e.g. [9, 11, 13]), but to the best of our knowledge, this is the first time when catalysts are used for tissue P systems[2]. Next, we provide the definition of tissue-like P systems with catalysts:

Definition 2.1. A *tissue-like P system with catalysts* 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 *catalytic rules* of the following form: $(cat \,|\, i, u/v, j)$ for $i, j \in \{0, 1, 2, \dots, q\}, i \neq j$ and $cat, u, v \in \Gamma^*$. The length of a catalytic rule is defined as $|u| + |v|$. The catalyst *cat* is not modified by the application of the rules.

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, i.e., as usual in tissue-like P systems, the edges linking cells are not provided explicitly: If a rule $(cat \,|\, i, u/v, j)$ is given, then cells i and j are considered linked.

The application of a catalytic rule $(cat \,|\, i, u/v, j)$ consists of the trade of the multiset u (initially in the cell i) against the multiset v (initially in j). The rule can be applied if the multiset *cat* is placed in the membrane with label i. We can also trade objects between one cell and the environment, labeled by 0. The rule is applied if in the cell with label i the objects of the multiset *cat* are present. If the catalyst *cat* is empty, then the rule is called a *communication rule.*

In our definition, all objects in the alphabet can act as catalysts, depending on the applied 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

[2] Comprehensive information about catalytic P systems can be found at [8].

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.

A *configuration* is an instantaneous description of the P system and it is represented as a tuple $(w'_1, \dots, w'_q)$. 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.

Example 2.1. Let us consider the following *tissue-like P system with catalysts* of degree 3, $\Pi = (\Gamma, \mathcal{E}, w_1, w_2, w_3, \mathcal{R}, i_0)$ where $\Gamma = \{a, b, c, d, e\}$, $\mathcal{E} = \{a, b\}$, $w_1 = a$, $w_2 = bd$ and $w_3 = ce$. The set $\mathcal{R}$ has four rules:

$$r_1 \equiv (c\,|\,2, b/a, 1) \quad r_2 \equiv (e\,|\,3, c/d, 2) \quad r_3 \equiv (3, d/a, 0) \quad r_4 \equiv (3, d/b^2, 0)$$

The output region is $i_0 = 3$. According to the set of rules, cell 2 is linked to cells 1 and 3 and cell 3 is linked to the environment. The computation starts from the initial configuration $C_0 = (w_1, w_2, w_3)$. Rule r_1 cannot be applied in this initial configuration, since the catalyst c does not appear in cell 2. Rules r_3 and r_4 cannot be applied, since object d is not placed in cell 3. In this initial configuration, we only can apply rule 2, since objects e and c are placed in membrane 3 and d is placed in membrane 2. After applying this rule, we obtain a new configuration. $C_1 = (w'_1, w'_2, w'_3)$ with $w'_1 = a$, $w'_2 = bc$ and $w'_3 = de$. Now, rule 1 can be applied by interchanging objects a and b, because the catalyst c is placed in cell 2. The communication rules 3 and 4 can be applied, since d appears in cell 3 and the environment contains objects a and b, but only one of them can be applied, and it is non-deterministically chosen. If we choose rule 3, then the new configuration is $C_2 = (w''_1, w''_2, w''_3)$ with $w''_1 = b$, $w''_2 = ac$ and $w''_3 = ae$. No more rules can be applied and the computation finishes. The output of the computation is the multiset in cell 3 in the halting configuration $w''_3 = ae$. If we choose rule 4, the obtained configuration is the same, but $w''_3 = b^2e$.

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

As pointed out above, given a binary 2D digital image, the HGB2I problem consists of calculating the number of connected components and the representative curves of the holes of these components (a hole is a white connected component surrounded by one black connected component). We can divide this problem in two sub-problems: the H_0 *problem* consists of calculating the number of connected components and the H_1 *problem*, which consists of calculating the number of holes. In this paper we present a new technique to use tissue-like P systems with catalysts to obtain homological information of binary 2D digital images.

A 2D digital image I can be considered as a matrix of objects called *pixels.* In this paper, a pixel of a digital image will be an object K_{ij}, where (i, j) denotes the position of the object in the matrix and K is the *color* of the pixel, $K = b$ (black) or $K = w$ (white).

The basic relation among pixels is the *adjacency* relation. There exist two natural possibilities: the 4-adjacency (Von Neumann neighborhood in cellular automata) and the 8-adjacency (Moore neighborhood in cellular automata).

In the first case, given a pixel K_{ij}, the set of its adjacent pixels is $\{K_{ij-1}, K_{ij+1}, K_{i-1j}, K_{i+1j}\}$, i.e., the adjacent pixels to any pixel K_{ij} are just north, south, west, east from it. In the second case, the set of adjacent pixels to K_{ij} 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 pixel are north, south, west, east ones and the diagonal pixels.

In this paper, we will consider 4-adjacency for black pixels and the 8-adjacency for white pixels. We must remember this consideration is due to the concept of holes. If we see Figure 1, we find two connected black components and no holes. If we would consider 4-adjacency for white pixels we would have a hole, then we choose 8-adjacency for white pixels.

3.1. Solving H_0 Problem

In order to provide a logarithmic-time uniform solution to the H_0 problem, we design a family of tissue-like P systems with catalysts, $\mathbf{\Pi_0} = \{\Pi_0(n)\}_{n\in\mathbb{N}}$. Each P system $\Pi_0(n)$ of the family will process all pictures of $n \times n$ pixels[3]. Given an $n \times n$ image I, we will consider a set of n^2 objects z_{ij} with $z = w$ or $z = b$ as the encoding of the image. Besides the objects w_{ij} and b_{ij}, the alphabet of the P system will contain objects $\{a_i : 1 \le i \le n+2\}$ which will be used as a counter and $\{(b_{ij}, (k,l)) : (1,1) \le (i,j) \le (k,l) \le (n,n)\}$, where we consider the lexicographic[4] order between pixels. Given an object $(b_{ij}, (k,l))$ we will say that (k,l) is a *label* associated with b_{ij}. We will also consider objects $\{A_{ijkl} : (1,1) \le (i,j) < (k,l) \le (n,n)\}$. The family of P systems is defined as follows:

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

where:

- $\Gamma = \{a_i : 1 \le i \le n+2\} \cup \{b_{ij}, w_{ij} : 1 \le i,j \le n\} \cup \{(b_{ij}, (k,l)) : (1,1) \le (i,j) \le (k,l) \le (n,n)\} \cup \{A_{ijkl} : (1,1) \le (i,j) < (k,l) \le (n,n)\}$.
- $\mathcal{E} = \Gamma - \{b_{ij}, w_{ij} : 1 \le i,j \le n\}$
- $\omega_1 = \{a_1\}$.
- $\omega_2 = \emptyset$.
- $\mathcal{R}$ is the set of rules:

[3] The input will be a squared digital image. If the initial image is not squared, it will be filled with white pixels.

[4] $(i,j) < (k,l) \Leftrightarrow i < k \vee (i = k \wedge j < l)$

- ◦ $R_1 \equiv (1, a_i/a_{i+1}, 0)$ for $1 \leq i \leq n+1$.
  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 positions in the array.
- ◦ $R_4 \equiv (1, (b_{ij}, (k,l))(b_{i'j'}, (k',l'))/(b_{ij}, (k',l'))(b_{i'j'}, (k',l'))A_{k'l'kl}, 0)$ for $(1,1) \leq (k',l') < (k,l) \leq (n,n)$ and (i,j), (i',j') adjacent positions in the array.
  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, objects A_{ijkl} act as catalysts. 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.

- $i_0 = 2$ is the output cell.

The initial configuration will be $\mathcal{C}_0 = (w_1 \cup I, w_2)$, where I is the encoding of the image by objects w_{ij} and b_{ij}.

Each system of the family implements the following stages:

1. *Label Allocation Stage*: Cell 1 trades objects b_{ij} against others of 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., $(1, (b_{ij}, (i',j'))(b_{kl}, (k',l'))/(b_{ij}, (i',j'))(b_{kl}, (i',j'))A_{i'j'k'l'}, 0)$, 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 traded against 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 of the form $(b_{ij}, (i,j))$ are sent to the output cell representing each one to a black connected component. The P system has 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 C_0 of the picture.

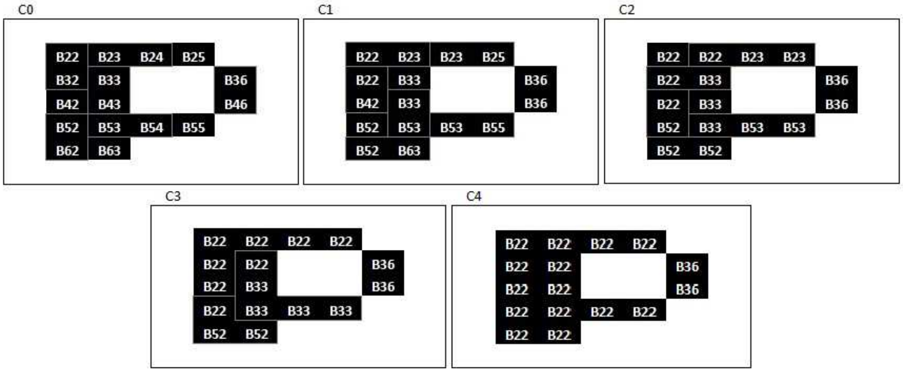


Fig. 1. A simple example to obtain H_0.

3.2. Solving H_1 Problem

With respect to the H_1 problem we use the same technique presented above, where labels are associated with the objects codifying pixels. The main technical novelty is that we introduce priorities in the rules of the tissue P system with catalysts. We construct a family of tissue-like P systems with catalysts $\mathbf{\Pi}_1 = \{\Pi_1(n)\}_{n\in\mathbb{N}}$ 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 the 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 $z = b \vee z = w$. The family of P systems is defined as follows:

$$\Pi_1(n) = (\Gamma, \mathcal{E}, \omega_1, \omega_2, \mathcal{R} = R_1 \cup \cdots \cup R_{10}\}, \{R_6, \mathcal{R}_8\} > R_1, i_0)$$

where:

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

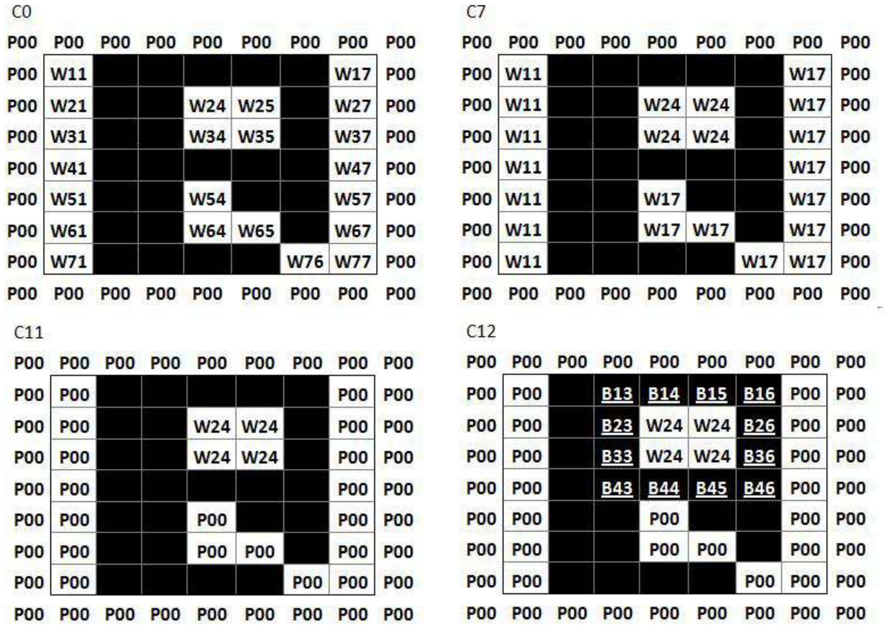
start the *Deleting Stage* after $n+2$ steps, and the *Segmenting Stage* after $n+4$ steps.

- ○ $R_2 \equiv (1, w_{ij}/(w_{ij},(i,j)), 0)$ for $1 \leq i,j \leq n$.
  These are the unique 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 $(1,1) \leq (k',l') < (k,l) \leq (n,n)$, $w_{ij}, w_{i'j'}$ 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} increases the speed of the process. 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 uses objects $(p_{ij},(0,0))$. We will refer to these objects as *pink* pixels. The rules are used in *Deleting Stage* to delete white pixels which are out of the connected black component. By using 8-adjacency, we transform outer white pixels into pink pixels, in order to differentiate them from the interior white pixels (holes).
- ○ $R_7 \equiv ((Z_{00ij}|1,(w_{i'j'},(i,j))/(p_{i'j'},(0,0)), 0)$.
  A new catalyst acts in the same way, trading white exterior pixel for pink pixels. In this way, the *Deleting Stage* takes only 2 steps.
- ○ $R_8 \equiv (z_{n+5}|1,(w_{ij},(i',j'))b_{kl}/(w_{ij},(i',j'))\bar{b}_{kl}, 0)$ 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 as well, so these rules send one white pixel for each hole to membrane number 2.

- $\{R_6, R_8\} > R_1$ represents the priority relation among the rules from the sets R_1,R_6 and R_1: Rules from sets R_6 and R_8 are applied before rules from the set R_1.
- $i_0 = 2$ is the output cell.

The initial configuration is $C_0 = (w_1 \cup I, w_2)$. 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 labels 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
$$(1, (w_{ij}, (i', j'))(w_{kl}, (k', l'))/(w_{ij}, (i', j'))(w_{kl}, (i', j'))Z_{i'j'k'l'}, 0)$$
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.



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.

3. *Deleting Stage*: Initially, the 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 associated label $(0, 0)$. 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 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, i.e., 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 Imagery. In this paper we provide a solution in the framework of tissue-like 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.

Acknowledgements. 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., Oswald M., *Tissue P Systems with Antiport Rules and Small Numbers of Symbols and Cells*, in *Developments in Language Theory* (2005), C. de Felice and A. Restivo, Eds., vol. **3572** of *Lecture Notes in Computer Science*, Springer, pp. 100–111.

[2] Bernardini F., Gheorghe M., *Cell Communication in Tissue P Systems: Universality Results*, Soft Computing, **9**, 9 (2005), pp. 640–649.

[3] Ceterchi R., Gramatovici R., Jonoska N., Subramanian K. G., *Tissue-like P Systems with Active Membranes for Picture Generation*, Fundamenta Informaticae, **56**, 4 (2003), pp. 311–328.

[4] Ceterchi R., Mutyam M., Păun Gh., Subramanian K. G., *Array-rewriting P Systems*, Natural Computing, **2**, 3 (2003), pp. 229–249.

[5] Chao J., Nakayama J., *Cubical Singular Simplex Model for 3D Objects and Fast Computation of Homology Groups*, in *13th International Conference on Pattern Recognition (ICPR'96)* (Los Alamitos, CA, USA, 1996), vol. **IV**, IEEE Computer Society, pp. 190–194.

[6] Christinal H. A., Díaz-Pernil D., Real P., *Segmentation in 2D and 3D Image Using Tissue-like P Systems*, in *CIARP* (2009), E. Bayro-Corrochano and J.-O. Eklundh, Eds., vol. **5856** of *Lecture Notes in Computer Science*, Springer, pp. 169–176.

[7] Christinal H. A., Díaz-Pernil D., Real P., *Using Membrane Computing for Obtaining Homology Groups of Binary 2D Digital Images*, in *IWCIA* (2009), P. Wiederhold and R. P. Barneva, Eds., vol. **5852** of *Lecture Notes in Computer Science*, Springer, pp. 383–396.

[8] Freund R., Ibarra O. H., *Catalytic P Systems*, in Păun et al. [20], 2010, ch. 4, pp. 83–117.

[9] Freund R., Kari L., Oswald M., Sosík P., *Computationally Universal P Systems without Priorities: Two Catalysts are Sufficient*, Theoretical Computer Science, **330**, 2 (2005), pp. 251–266.

[10] Freund R., Păun Gh., Pérez-Jiménez M. J., *Tissue P Systems with Channel States*, Theoretical Computer Science, **330**, 1 (2005), pp. 101–116.

[11] Krishna S. N., *On Pure Catalytic P Systems*, in *UC* (2006), C. S. Calude, M. J. Dinneen, Gh. Păun, G. Rozenberg, and S. Stepney, Eds., vol. **4135** of *Lecture Notes in Computer Science*, Springer, pp. 152–165.

[12] Krishna S. N., Lakshmanan K., Rama R., *Tissue P Systems with Contextual and Rewriting Rules*, in *WMC-CdeA* (2002), Gh. Păun, G. Rozenberg, A. Salomaa, and C. Zandron, Eds., vol. **2597** of *Lecture Notes in Computer Science*, Springer, pp. 339–351.

[13] Krishna S. N., Păun A., *Results on Catalytic and Evolution-communication P Systems*, New Generation Computing, **22**, 4 (2004).

[14] Martín-Vide C., Pazos J., Păun Gh., Rodríguez-Patón A., *A New Class of Symbolic Abstract Neural Nets: Tissue P Systems*, in *COCOON* (2002), O. H. Ibarra and L. Zhang, Eds., vol. **2387** of *Lecture Notes in Computer Science*, Springer, pp. 290–299.

[15] Martín-Vide C., Păun Gh., Pazos J., Rodríguez-Patón A., *Tissue P Systems*, Theoretical Computer Science, **296**, 2 (2003), pp. 295–326.

[16] Păun A., Păun Gh., *The Power of Communication: P Systems with Symport/Antiport*, New Generation Computing, **20**, 3 (2002), pp. 295–306.

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

[18] Păun Gh., *Membrane Computing. An Introduction.*, Springer-Verlag, Berlin, Germany, 2002.

[19] Păun Gh., Pérez-Jiménez M., Riscos-Núñez A., *Tissue P Systems with Cell Division*, International Journal of Computers, Communication and Control, **3**, 3 (2008), pp. 295–303.

[20] Păun Gh., Rozenberg G., Salomaa A., Eds., *The Oxford Handbook of Membrane Computing.* Oxford University Press, 2010.

[21] Peltier S., Ion A., Haxhimusa Y., Kropatsch W. G., Damiand G., *Computing Homology Group Generators of Images Using Irregular Graph Pyramids*, in *GbRPR* (2007), F. Escolano and M. Vento, Eds., vol. **4538** of *Lecture Notes in Computer Science*, Springer, pp. 283–294.

[22] P system web page: `http://ppage.psystems.eu/`