A path to computational efficiency through membrane computing

David Orellana-Martín, Luis Valencia-Cabrera, Agustín Riscos-Núñez, Mario J. Pérez-Jiménez

Research Group on Natural Computing, Department of Computer Science and Artificial Intelligence, Universidad de Sevilla, Avda. Reina Mercedes s/n, 41012, Sevilla, Spain

Keywords:
Membrane computing
P system with symport/antiport
Membrane division
Computational complexity

ABSTRACT

The search for new mechanisms and tools allowing us to tackle the famous **P** versus **NP** problem from new perspectives is an important task, due to the relevance of that problem. The concept of efficiency of computing models is associated with the ability to solve intractable (in a classical sense) problems in polynomial time. Assuming that **P** $\neq$ **NP**, that concept is equivalent to the capability to solve **NP**-complete problems in an efficient way. Different frontiers of the efficiency have been given in Membrane Computing in terms of syntactical or semantic ingredients of the models. In particular, in the framework of tissue P systems with cell division using symport/antiport rules, the length of communication rules (passing from length 1 to length 2) provides an optimal borderline of the efficiency.

Cell-like P systems with symport/antiport rules and membrane division is a restricted variant of such tissue P systems in both its structure (rooted tree versus undirected graph) and in the way membranes communicate with each other and with the environment. The limitations of efficient computations in such kind of P systems which use non-cooperative communication rules have been previously established. In this paper, a uniform polynomial time solution for the *Hamiltonian cycle problem*, a well known **NP**-complete problem, by means of cell-like P systems with membrane division using minimal cooperation in communication rules (at most two objects are involved), is provided. Hence, a new optimal boundary between tractability and **NP**-hardness, is provided: passing from non-cooperative rules to cooperative rules in cell-like P systems with symport/antiport rules and membrane division amounts to passing from non-efficiency to efficiency.

1. Introduction

The **P** versus **NP** problem is one of the most important open problems that have been formulated in theoretical computer science. This is the first problem of the famous list of the *Millenium Prize Problems*, including the seven greatest unsolved mathematical problems. Solving this problem might produce a huge economic impact. In an informal way, we can say that this problem analyzes whether or not *finding solutions* is harder than *checking the correctness* of possible solutions. It is widely believed that it is harder *to solve* a problem than *to check* that a solution is valid/good; that is, it is conjectured that

E-mail addresses: dorellana@us.es (D. Orellana-Martín), lvalencia@us.es (L. Valencia-Cabrera), ariscosn@us.es (A. Riscos-Núñez), marper@us.es (M.J. Pérez-Jiménez).

$\mathbf{P} \neq \mathbf{NP}$. The classical approach to solve this conjecture consists of considering a single **NP**-complete problem and trying to prove whether that problem belongs to the class **P** or not. In computing models, new tools to tackle the **P** versus **NP** problem can be provided in terms of syntactical or semantic ingredients in the model. In this context, it is interesting to determine ingredients that allow us to efficiently solve computationally hard problems, and then investigate which ones should be removed in order to reduce the power so that only problems from **P** can be efficiently solved. The use (or not) of such ingredients provides a frontier between tractability and **NP**-hardness.

Membrane computing is a flexible and versatile branch of natural computing, which arises as an abstraction of the compartmentalized structure of living cells, and the way biochemical substances are processed in (or moved between) membrane-bounded regions [18]. Inspired by the structure of living cells, two main classes of membrane systems have been investigated: a hierarchical (cell-like) arrangement of membranes, inspired from the structure of the cell [18] and a network of membranes (placed in the nodes of a directed graph), inspired from the cell-interconnection in tissues [13] or inspired from the way that neurons communicate with each other by means of short electrical impulses (spikes), emitted at precise moments of time [6]. All classes of computing devices considered in the field of membrane computing are generally called *P systems*, which are parallel and distributed computational models based on processing multisets of objects in cell-like or tissue-like structures by means of rewriting rules. A P system is *cooperative* if it contains rules that need more than one object to be triggered. A comprehensive information in membrane computing can be found in [20] and [22], and for the most up-to-date source of this area, please refer to [30].

Cell-like P systems with symport/antiport rules were introduced in [16], aiming to abstract the biological phenomenon of trans-membrane transport of pairs of chemical substances, in the same or in opposite directions. On the other hand, tissue P systems with symport/antiport rules were introduced in [17] by abstracting networks of elementary cells such that some of them are linked by "communication channels".

In eukaryotic cells, there are two relevant processes: *mitosis* and *membrane fission*. The first one is a process of nuclear division in eukaryotic cells during which one cell gives rise to two genetically identical children cells. *Membrane fission* occurs when a membrane leads to two separated membranes; that is, whenever a vesicle is produced or a larger subcellular compartment is divided into smaller discrete units. These processes have been a source of inspiration to incorporate new syntactical ingredients in membrane computing in order to be able to produce exponential workspace (by means of compartments) in polynomial — often linear — time. Specifically, inspired by the mitosis process, *membrane division* rules were defined in the framework of cell-like P systems providing computing devices called *P systems with active membranes* [19]. With respect to the membrane fission process, *P systems with membrane separation* were introduced in [14]. These concepts were also considered in the framework of tissue-like P systems: *tissue P systems with cell division* [21] and *tissue P systems with cell separation* [15].

In previous works [4,5,8,7,15,23,25,26], new tools to tackle the **P** versus **NP** problem are given in the framework of *Membrane Computing*. Let us recall that using families of non-cooperative tissue P systems with symport rules involving only one object and cell division, only problems in the complexity class **P** can be solved in polynomial time; that is $\mathbf{P} = \mathbf{PMC_{TDC(1)}}$ [3]. Nevertheless, families of tissue P systems with cell division and symport/antiport rules involving at most two objects (*minimal cooperation*) can solve **NP**-complete problems in polynomial time, which implies $\mathbf{NP} \cup \mathbf{co} - \mathbf{NP} \subseteq \mathbf{PMC_{TDC(2)}}$ [29]. Hence, *cooperation in communication rules* provide a frontier of the efficiency in the framework of tissue P systems with symport/antiport rules and cell division, assuming that $\mathbf{P} \neq \mathbf{NP}$.

This paper deals with cell-like *P systems with symport/antiport rules and membrane division* whose processor units are called *membranes*. This kind of P systems have important differences with respect to the tissue-like approach, where processor units are called *cells*: (a) the underlying structure is a rooted tree instead of a directed graph; (b) the rules are associated with the membranes instead of with the whole system; (c) the communication is only produced between a parent membrane and one of its child membranes instead of any two arbitrary cells; (d) only the skin membrane, the most external one, can communicate with the environment instead of each cell; and (e) only elementary membranes can be divided instead of each arbitrary cell. By using families of recognizer cell-like P systems with membrane division which use communication rules with length at most 3, uniform polynomial-time solutions to **NP**-complete problems have been provided [12]. In this paper, the previous result is improved showing that minimal cooperation (at most two objects involved in communication rules) is enough in order to solve **NP**-complete problems in uniform polynomial-time. Besides, it is known that with non-cooperative rules, only problems in the class **P** can be solved efficiently [10]. In this manner, similar complexity results that in tissue-like P systems with symport/antiport rules and cell division have been obtained, but in a more restrictive syntactical context.

The paper is structured as follows. In Section 2, some concepts needed for a more comprehensive reading of the paper are presented. The framework of recognizer cell-like P systems with symport/antiport rules and membrane division is introduced at Section 3. A uniform polynomial-time solution of the `HAM-CYCLE` problem by means of a family of P systems with membrane division using symport/antiport rules of length at most 2, is described in Section 4. Conclusions and some open problems are formulated in the last section.

2. Preliminaries

In order to provide a self-contained paper, some basic concepts and notations are introduced in this section.

2.1. Languages and multisets

An *alphabet* Γ is a non-empty set and its elements are called *symbols*. A *string* u over Γ is a mapping from a natural number $n \in \mathbb{N}$ onto Γ. Number n is called *length* of the string u and it is denoted by $|u|$. The empty string (with length 0) is denoted by λ. A *language* over Γ is a set of strings over Γ.

A *multiset* over an alphabet Γ is an ordered pair (Γ, f), where f is a mapping from Γ onto the set of natural numbers $\mathbb{N}$. For each $x \in \Gamma$ we say that $f(x)$ is the *multiplicity* of x in that multiset. The *support* of a multiset $m = (\Gamma, f)$ is defined as $supp(m) = \{x \in \Gamma \mid f(x) > 0\}$. A multiset is finite if its support is a finite set. We denote by $\emptyset$ the empty multiset. The size of a finite multiset $m = (\Gamma, f)$ is $\sum_{x\in\Gamma} f(x)$, and it is denoted by $|m|$. Let $m_1 = (\Gamma, f_1)$, $m_2 = (\Gamma, f_2)$ be multisets over Γ, then the union of m_1 and m_2, denoted by $m_1 + m_2$, is the multiset (Γ, g), where $g(x) = f_1(x) + f_2(x)$ for each $x \in \Gamma$. We say that m_1 is contained in m_2 and we denote it by $m_1 \subseteq m_2$, if $f_1(x) \le f_2(x)$ for each $x \in \Gamma$. The relative complement of m_2 in m_1, denoted by $m_1 \setminus m_2$, is the multiset (Γ, g), where $g(x) = f_1(x) - f_2(x)$ if $f_1(x) \ge f_2(x)$, and $g(x) = 0$ otherwise. Let us note that a set is a particular case of a multiset when each symbol of the support has multiplicity 1.

Notation: given a sequence $x_1, \dots, x_n$ of symbols of an alphabet, we denote by $x_1, \dots, \widehat{x_k}, \dots, x_n$ the sequence $x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n$; that is, the k-th term, x_k, does not appear in that sequence.

2.2. Graphs and Hamiltonian cycles

We recall some concepts from graph theory that are used in this paper. For more details on graph theory, please refer to [1].

A *rooted tree* is a connected, acyclic, undirected graph in which one of the vertices (called *the root of the tree*) is distinguished from the others. Given a node x (different from the root) in a rooted tree, if the last edge on the (unique) path from the root to node x is $\{x, y\}$ (so $x \neq y$), then y is the *parent* of node x, and x is a *child* of node y. We denote it by $y = p(x)$ and $x \in ch(y)$. The root is the only node in the tree with no parent. A node with no children is called a *leaf*.

Let $G = (V, E)$ be a directed graph, where the set of nodes is V and the set of arcs is $E \subseteq V \times V$. We say that a finite sequence $\gamma = (x_{i_1}, x_{i_2}, \dots, x_{i_r}, x_{i_{r+1}})$ of nodes of G is a *simple path* of G from x_{i_1} to $x_{i_{r+1}}$ of length $r \ge 1$ if $(x_{i_j}, x_{i_{j+1}}) \in E$ for each $1 \le j \le r$, and $x_{i_j} \neq x_{i_{j'}}$ for each pair $1 \le j < j' \le r+1$. A finite sequence $\gamma = (x_{i_1}, x_{i_2}, \dots, x_{i_r}, x_{i_{r+1}})$ with $x_{i_{r+1}} = x_{i_1}$ is a *simple cycle* of length r, if the sequence $(x_{i_1}, x_{i_2}, \dots, x_{i_r})$ is a simple path and $(x_{i_r}, x_{i_{r+1}}) \in E$. A *Hamiltonian cycle* of G is a simple cycle $\gamma = (u_{\alpha_1}, u_{\alpha_2}, \dots, u_{\alpha_r}, u_{\alpha_1})$ of G such that $V = \{u_{\alpha_1}, u_{\alpha_2}, \dots, u_{\alpha_r}\}$.

Let $(x_{i_1}, x_{i_2}, \dots, x_{i_r}, x_{i_{r+1}})$ be a simple path of a directed graph $G = (V, E)$ with $V = \{1, \dots, n\}$. It is also denoted by the set $\{(x_{i_1}, x_{i_2})_1, \dots, (x_{i_r}, x_{i_{r+1}})_r\}$. That is, the labelled arc $(x_{i_k}, x_{i_{k+1}})_k$ can be interpreted as the k-th arc of the path γ, for each k $(1 \le k \le r)$. Throughout this work we denote

$$\begin{aligned}
A_G &= \{(i, j)_k \mid 1 \le i, j, k \le n, (i, j) \in E\};\\
A'_G &= \{(i, j)'_k \mid 1 \le i, j, k \le n, (i, j) \in E\};\\
A''_G &= \{(i, j)''_k \mid 1 \le i, j, k \le n, (i, j) \in E\}.
\end{aligned}$$

It is easy to prove the following result:

Proposition 2.1. *Let $G = (V, E)$ be a directed graph, $V = \{1, \dots, n\}$ and $A_G = \{(i, j)_k \mid 1 \le i, j, k \le n, (i, j) \in E\}$. If $B \subseteq A_G$, then the following assertions are equivalent:*

(1) B is a Hamiltonian cycle.
(2) $|B| = n$ and the following holds: for each $i, i', j, j', k, k' \in \{1, \dots, n\}$,
  (a) if $(i, j)_k \in B$, $(i', j')_{k'} \in B$, and $(i, j)_k \neq (i', j')_{k'}$, then $k \neq k'$;
  (b) if $(i, j)_k \in B$, $(i', j')_{k'} \in B$, and $(i, j)_k \neq (i', j')_{k'}$, then $i \neq i'$;
  (c) if $(i, j)_k \in B$, $(i', j')_{k'} \in B$, and $(i, j)_k \neq (i', j')_{k'}$, then $j \neq j'$;
  (d) if $(i, j)_k \in B$, and $(i', j')_{k+1} \in B$, then $j = i'$.

As a consequence, the following are true:

Remark 1. Let $B \subseteq A_G$ be a Hamiltonian cycle of G. For each $i, i', j, j', k, k' \in \{1, \dots, n\}$, the following holds:

- if $(i, j)_k \in B$ and $j \neq j'$, then $(i, j')_{k'} \notin B$;
- if $(i, j)_k \in B$ and $i \neq i'$, then $(i', j)_{k'} \notin B$;
- if $(i, j)_k \in B$ and $(i, j) \neq (i', j')$, then $(i', j')_k \notin B$;
- if $(i, j)_k \in B$ and $j \neq i'$, then $(i', j')_{k+1} \notin B$.

Remark 2. If $(x_{i_1}, x_{i_2}, \dots, x_{i_n}, x_{i_1})$ is a Hamiltonian cycle of G, then we can describe it by the set $B_1 = \{(x_{i_1}, x_{i_2})_1, (x_{i_2}, x_{i_3})_2, \dots, (x_{i_n}, x_{i_1})_n\} \subseteq A_G$. But $(x_{i_2}, x_{i_3}, \dots, x_{i_n}, x_{i_1}, x_{i_2})$ also represents the same Hamiltonian cycle and can be described as $B_2 = \{(x_{i_2}, x_{i_3})_1, (x_{i_3}, x_{i_4})_2, \dots, (x_{i_1}, x_{i_2})_n\}$, and so on up to B_n. Thus, given a Hamiltonian cycle of G, there are exactly n different subsets of A_G codifying that cycle.

Remark 3. Let us suppose that the total number of Hamiltonian cycles of G is q. Then, the number of different subsets B of A_G verifying conditions (a), (b), (c), and (d) from Proposition 2.1 is exactly $n \cdot q$.

2.3. Decision problems and languages

Roughly speaking, a decision problem X is one whose solution/answer is either "yes" or "no". This can be formally defined by an ordered pair (I_X, θ_X), where I_X is a language over a finite alphabet Σ_X and θ_X is a total Boolean function over I_X. The elements of I_X are called *instances* of the problem X. Each decision problem X has associated a language L_X over the alphabet Σ_X as follows: $L_X = \{u \in I_X \mid \theta_X(u) = 1\}$; that is, L_X is the set of inputs for which the answer is affirmative. Conversely, every language L over an alphabet Σ has associated a decision problem $X_L = (I_{X_L}, \theta_{X_L})$ as follows: $I_{X_L} = \Sigma^*$ and $\theta_{X_L}(u) = 1$ if and only if $u \in L$. Then, given a decision problem X we have $X_{L_X} = X$, and given a language L over an alphabet Σ we have $L_{X_L} = L$.

The complement problem $\overline{X}$ of a decision problem $X = (I_X, \theta_X)$ is the decision problem $(I_X, \neg\theta_X)$; that is, $L_{\overline{X}} = \Sigma_X^* \setminus L_X$: for each instance the answer of $\overline{X}$ is "yes" if and only if the answer of X is "no".

2.4. The Cantor pairing function

The Cantor pairing function encodes pairs of natural numbers by single natural numbers and it is defined as follows: for each $m, n \in \mathbb{N}$

$$\langle m, n \rangle = \frac{(m+n)(m+n+1)}{2} + m$$

The Cantor pairing function is a primitive recursive bijective function from $\mathbb{N} \times \mathbb{N}$ onto $\mathbb{N}$. Then, for each $t \in \mathbb{N}$ there exists a unique natural number $m, n \in \mathbb{N}$ such that $t = \langle m, n \rangle$.

3. P systems with symport/antiport rules and division rules

A kind of cell-like P systems that use communication rules capturing the biological phenomenon of trans-membrane transport of several chemical substances was introduced in [16]. Specifically, two processes were considered. The first one allows a multiset of chemical substances to pass through a membrane in the same direction. In the second one, two multisets of chemical substances (located in different biological membranes) only pass with the help of each other (an *exchange* of objects between both membranes). This is the computing framework where this paper is developed. Next, recognizer P systems with symport/antiport rules and membrane division are briefly introduced (see [9] for more details).

Recognizer membrane systems were introduced in [28], and they provide a natural framework to solve decision problems by means of families of membrane systems. We adapt this definition to P systems with symport/antiport rules and membrane division in a natural way.

Definition 3.1. A recognizer P system with symport/antiport rules and membrane division of degree $q \geq 1$ is a tuple

$$\Pi = (\Gamma, \mathcal{E}, \Sigma, \mu, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}_1, \dots, \mathcal{R}_q, i_{in}, i_{out})$$

where:

- $\Gamma, \mathcal{E}, \Sigma$ are finite alphabets such that Γ has two distinguished symbols yes and no, $\mathcal{E} \subsetneq \Gamma$, $\Sigma \subsetneq \Gamma$ and $\mathcal{E} \subseteq \Gamma \setminus \Sigma$.
- μ is a rooted tree whose nodes are injectively labelled with $1, \dots, q$ (the root of the tree is labelled with *1)*.
- $\mathcal{M}_1, \dots, \mathcal{M}_q$ are finite multisets over $\Gamma \setminus \Sigma$ such that at least one copy of yes and no is present in some of them.
- $\mathcal{R}_i$, $1 \leq i \leq q$, are finite sets of rules over Γ of the following forms:
  - ∗ *Communication rules*:
    - ⋆ *Symport rules*: (u, out) or (u, in), where u is a multiset over Γ such that $|u| > 0$.
    - ⋆ *Antiport rules*: $(u, out; v, in)$, where u, v are multisets over Γ such that $|u| > 0$ and $|v| > 0$.
  - ∗ *Division rules*: $[\, a \,]_i \to [\, b \,]_i \, [\, c \,]_i$, where $a, b, c \in \Gamma$, $i \in \{2, \dots, q\}$, $i \neq i_{out}$, and i is the label of a leaf of the tree μ.
- $i_{in} \in \{1, \dots, q\}$ and $i_{out} = 0$ *(0* is the label of the environment*)*.
- All computations halt.
- If $\mathcal{C}$ is a computation of Π, then either symbol yes or symbol no (but not both) must have been released into the environment, and only at the last step of the computation.

For each multiset m over Σ, the initial configuration of Π *with input multiset* m is $\mathcal{M}_1, \dots, \mathcal{M}_{i_{in}} + m, \dots, \mathcal{M}_q$, that is, the input multiset m is added to the contents of the input membrane. We denote by $\Pi + m$ the system Π with input multiset m.

A *computation* of Π is a finite sequence of configurations such that the first term of the sequence is the initial configuration of the system and each non-first term of the sequence from the previous configuration by applying rules of the system in a non-deterministic maximally parallel manner; that is, no more rules could be applied to such configuration. If no rules can be applied to a configuration, we say that it is a *halting configuration* and the system finishes the computation. We say that a computation $\mathcal{C}$ of Π is an *accepting computation* (respectively, *rejecting computation*) if object $\texttt{yes}$ (respectively, object $\texttt{no}$) appears in the environment associated with the corresponding halting configuration of $\mathcal{C}$, and neither object $\texttt{yes}$ nor $\texttt{no}$ appears in the environment associated with any non-halting configuration of $\mathcal{C}$.

For each natural number $k \geq 1$, $\mathbf{CDC}(k)$ (respectively, $\mathbf{CDA}(k)$ or $\mathbf{CDS}(k)$) denotes the class of all recognizer P systems with symport/antiport rules (respectively, only antiport rules or only symport rules) and membrane division such that the length of the communication rules is at most k.

Next, the concept of efficient solvability for decision problems by means of families of recognizer membrane systems (see [24] for more details) is adapted to the new framework in a natural way.

Definition 3.2. A decision problem X is solvable in uniform polynomial-time by a family $\mathbf{\Pi} = \{\Pi(n) \mid n \in \mathbb{N}\}$ of recognizer P systems with symport/antiport rules and membrane division, if the following holds:

- the family $\mathbf{\Pi}$ is polynomially uniform by Turing machines; that is, there exists a deterministic Turing machine working in polynomial time which constructs the system $\Pi(n)$;
- there exists a pair (cod, s) of polynomial-time computable functions over I_X such that:
  - ⋆ for each instance $u \in I_X$, $s(u)$ is a natural number and $cod(u)$ is an input multiset of the system $\Pi(s(u))$;
  - ⋆ for each $n \in \mathbb{N}$, $s^{-1}(n)$ is a finite set;
  - ⋆ the family $\mathbf{\Pi}$ is polynomially bounded with regard to (X, cod, s);
  - ⋆ the family $\mathbf{\Pi}$ is sound and complete with regard to (X, cod, s)

The instance $u \in I_X$ will be processed by the system $\Pi(s(u))$ with input multiset $cod(u)$. Furthermore, from the soundness and completeness is deduced that $\Pi(s(u)) + cod(u)$ is a *confluent* system, in the sense that all possible computations of $\Pi(s(u)) + cod(u)$ must give the same answer.

If $\mathbf{R}$ is a class of recognizer P systems with symport/antiport rules and membrane division, then we denote by $\mathbf{PMC_R}$ the set of all decision problems which can be solved in uniform polynomial-time by means of recognizer P systems from $\mathbf{R}$. The class $\mathbf{PMC_R}$ is closed under complement and polynomial-time reductions (see [27] for details).

In [10], the limitations of efficient computations in recognizer P systems with symport/antiport rules and membrane division when communication rules are non-cooperative rules have been established. Specifically, in the cited paper it has been shown that $\mathbf{PMC_{CDC(1)}} = \mathbf{P}$. This result has been obtained by applying the dependency graph technique (a directed graph is associated with each P system Π verifying the following: there exists an accepting computation of Π if and only if there exists a path between two distinguished nodes in the dependency graph associated with it). From a complexity point of view, what happens if *minimal cooperation* is considered in communication rules? Do these systems have the ability to solve computationally hard problems in uniform polynomial-time?

4. On the presumed efficiency of CDC(2)

This section is devoted to justifying the presumed efficiency of recognizer P systems with membrane division which use *minimal cooperation* in communication rules. In [12], an efficient solution to the $\texttt{SAT}$ problem was given by using a family of systems from $\mathbf{CDC}(3)$. In this section, this result is improved by giving a uniform polynomial-time solution to the $\texttt{HAM-CYCLE}$ problem (*given a directed graph, to determine whether or not there exists a Hamiltonian cycle in the graph*), a well-known **NP**-complete problem [2], by means of a family of systems from $\mathbf{CDC}(2)$.

4.1. A uniform polynomial-time solution of the $\texttt{HAM-CYCLE}$ *problem in* $\mathbf{CDC}(2)$

For each $m, n \in \mathbb{N}$, being n the number of nodes and m the number of edges of a directed graph G, we consider the recognizer P system with symport/antiport rules and membrane division of degree $11 + 2n + n^3$

$$\begin{aligned}\Pi(\langle m, n\rangle) = (&\Gamma, \mathcal{E}, \Sigma, \mu, \mathcal{M}_r\,(1 \leq r \leq 11)\,,\, \mathcal{M}_{a_{1,j}}\,(1 \leq j \leq n),\\ &\mathcal{M}_{a_{2,j}}\,(1 \leq j \leq n)\,,\, \mathcal{M}_{e_{i,j,k}}(1 \leq i, j, k \leq n)\,,\, \mathcal{R}_r\,(1 \leq r \leq 11),\\ &\mathcal{R}_{a_{1,j}}\,(1 \leq j \leq n)\,,\, \mathcal{R}_{a_{2,j}}\,(1 \leq j \leq n)\,,\, \mathcal{R}_{e_{i,j,k}}(1 \leq i, j, k \leq n),\\ &i_{in}, i_{out})\end{aligned}$$

defined as follows (parameter m is involved in the definition of some rules):

- The working alphabet Γ is:

$$\begin{array}{l}\Sigma \cup \mathcal{E} \cup \{b'_r, b''_r, b'''_r, c'_r, c''_r, c'''_r, c''''_r \mid 1 \le r \le n^3\} \cup \\ \{\beta_r \mid 0 \le r \le n^3+7\} \cup \{(i,j)'_k,\ (i,j)''_k \mid 1 \le i,j,k \le n\} \cup \\ \{(i,j)''_{k,r} \mid 1 \le i,j,k \le n \wedge 1 \le r \le n^3\} \cup \\ \{\alpha_0, a, a', a'', b, b', b'', b''', c, c', c'', c''', c'''', \mathtt{yes}, \mathtt{no}\},\end{array}$$

  where the input alphabet is $\Sigma = \{(i,j)_k \mid 1 \le i,j,k \le n\}$, and the alphabet of the environment is $\mathcal{E} = \{\alpha_r \mid 1 \le r \le n^3+6\}$
- Membrane structure μ: the set of nodes is

$$V = \{1, \dots, 11\} \cup \{a_{1,j}, a_{2,j} \mid 1 \le j \le n\} \cup \{e_{i,j,k} \mid 1 \le i,j,k \le n\}$$

  The root is labelled by 1 and the remaining nodes are its children.
- Initial multisets:

$$\begin{array}{l}\mathcal{M}_1 = \{\alpha_0\} \cup \{\beta_r \mid 1 \le r \le n^3+7\} \cup \\ \{b'_r, b''_r, b'''_r, c'_r, c''_r, c'''_r, c''''_r \mid 1 \le r \le n^3-1\};\ \mathcal{M}_2 = \{a^n, b, c\};\ \mathcal{M}_3 = \{b'_{n^3}\};\\ \mathcal{M}_4 = \{b''_{n^3}\};\ \mathcal{M}_5 = \{b'''_{n^3}\};\ \mathcal{M}_6 = \{c'_{n^3}\};\ \mathcal{M}_7 = \{c''_{n^3}\};\ \mathcal{M}_8 = \{c'''_{n^3}\};\\ \mathcal{M}_9 = \{c''''_{n^3}\};\ \mathcal{M}_{10} = \{\mathtt{yes}\};\ \mathcal{M}_{11} = \{\mathtt{no}, \beta_0\};\\ \mathcal{M}_{a_{1,j}} = \{a'_{n^3}\},\ \mathcal{M}_{a_{2,j}} = \{a''_{n^3}\}, \text{for } 1 \le j \le n;\\ \mathcal{M}_{e_{i,j,k}} = \{(i,j)''_{k,n^3}\}, \text{for } 1 \le i,j,k \le n.\end{array}$$

- Rules of the system:

  $\boxed{\text{Rules in } \mathcal{R}_1}$

  - Rules to control the output of the computations by counters α_r: $\{(\alpha_r, out; \alpha_{r+1}, in) \mid 0 \le r \le n^3+5\}$ (**1.1**)
  - Rule to produce an affirmative answer: $(\mathtt{yes}, out)$ (**1.2**)
  - Rule to produce a negative answer: $(\mathtt{no}\,\alpha_{n^3+6}, out)$ (**1.3**)

  $\boxed{\text{Rules in } \mathcal{R}_2}$

  - Rules to produce all possible subsets of A'_G in membranes labelled by 2 at configuration $\mathcal{C}_{n^3+1}$:
    $\{[\,(i,j)_k\,]_2 \to [\,(i,j)'_k\,]_2\ [\,\#\,]_2 \mid 1 \le i,j,k \le n\}$ (**2.1**)
  - Rules to move objects a', a'', b', b'', c''', c', c'', c''' and c'''' into membranes labelled by 2 at configurations $\mathcal{C}_{n^3+2}$, $\mathcal{C}_{n^3+3}$, $\mathcal{C}_{n^3+4}$ and $\mathcal{C}_{n^3+5}$, respectively.
    $(a, out; a', in)$; $(a', out; a'', in)$ (**2.2**)
    $(b, out; b', in)$; $(b', out; b'', in)$; $(b'', out; b''', in)$ (**2.3**)
    $(c, out; c', in)$; $(c', out; c'', in)$; $(c'', out; c''', in)$; $(c''', out; c'''', in)$ (**2.4**)
    $(a''\,b''', out)$; $(b'''\,c'''', out)$ (**2.5**)
  - Rules to produce in each membrane labelled by 2 at configuration $\mathcal{C}_{n^3+2}$ a subset of A''_G from a subset of A'_G at configuration $\mathcal{C}_{n^3+1}$:
    $\{((i,j)'_k, out; (i,j)''_k, in) \mid 1 \le i,j,k \le n\}$ (**2.6**)
  - Rules to generate in each membrane labelled by 2 at configuration $\mathcal{C}_{n^3+1}$ a subset of A''_G encoding a possible Hamiltonian cycle.

$$(\mathbf{2.7}) \left\{ \begin{array}{l} \{((i,j)''_k\ (i,j')''_{k'}, out) \mid 1 \le i,i',j,j',k,k' \le n\} \cup \\ \{((i,j)''_k\ (i',j)''_{k'}, out) \mid 1 \le i,i',j,j',k,k' \le n\} \cup \\ \{((i,j)''_k\ (i',j')''_{k+1}, out) \mid 1 \le i,i',j,j',k,k' \le n,\ j \ne i'\} \cup \\ \{((i,j)''_k\ (i',j')''_k, out) \mid 1 \le i,i',j,j',k,k' \le n\} \end{array} \right.$$

  - Rules to check if the subset represented by each membrane with label 2 at configuration $\mathcal{C}_{n^3+3}$ encodes a Hamiltonian cycle of the input graph: $\{(a''\,(i,j)''_k, out) \mid 1 \le i,j,k \le n\}$ (**2.8**)

  $\boxed{\text{Rules in } \mathcal{R}_3}$

  Rules to produce $2^{n \cdot m}$ copies of object b' in the skin membrane of configuration $\mathcal{C}_{n^3+1}$:
  $\{(b'_r, out; b'_{r-1}, in) \mid n \cdot m+1 \le r \le n^3\}$ (**3.1**)
  $\{[\,b'_r\,]_3 \to [\,b'_{r-1}\,]_3\ [\,b'_{r-1}\,]_3 \mid 2 \le r \le n \cdot m\}$ (**3.2**)
  $[\,b'_1\,]_3 \to [\,b'\,]_3\ [\,b'\,]_3$ (**3.3**)
  (b', out) (**3.4**)

  $\boxed{\text{Rules in } \mathcal{R}_4}$

  Rules to produce $2^{n \cdot m}$ copies of object b'' in the skin membrane at configuration $\mathcal{C}_{n^3+1}$:
  $\{(b''_r, out; b''_{r-1}, in) \mid n \cdot m+1 \le r \le n^3\}$ (**4.1**)
  $\{[\,b''_r\,]_4 \to [\,b''_{r-1}\,]_4\ [\,b''_{r-1}\,]_4 \mid 2 \le r \le n \cdot m\}$ (**4.2**)
  $[\,b''_1\,]_4 \to [\,b''\,]_4\ [\,b''\,]_4$ (**4.3**)
  (b'', out) (**4.4**)

$\boxed{\text{Rules in } \mathcal{R}_5}$

Rules to produce $2^{n\cdot m}$ copies of object b''' in the skin membrane at configuration $\mathcal{C}_{n^3+1}$:

$\{(b'''_r\,,\, out\,;\, b'''_{r-1}\,,\, in) \mid n\cdot m+1 \le r \le n^3\}$ (**5.1**)

$\{[\,b'''_r\,]_5 \rightarrow [\,b'''_{r-1}\,]_5\ [\,b'''_{r-1}\,]_5 \mid 2 \le r \le n\cdot m\}$ (**5.2**)

$[\,b'''_1\,]_5 \rightarrow [\,b'''\,]_5\ [\,b'''\,]_5$ (**5.3**)

$(b'''\ ,\ out)$ (**5.4**)

$\boxed{\text{Rules in } \mathcal{R}_6}$

Rules to produce $2^{n\cdot m}$ copies of object c' in the skin membrane at configuration $\mathcal{C}_{n^3+1}$:

$\{(c'_r\,,\, out\,;\, c'_{r-1}\,,\, in) \mid\ n\cdot m+1 \le r \le n^3\}$ (**6.1**)

$\{[\,c'_r\,]_6 \rightarrow [\,c'_{r-1}\,]_6\ [\,c'_{r-1}\,]_6 \mid 2 \le r \le n\cdot m\}$ (**6.2**)

$[\,c'_1\,]_6 \rightarrow [\,c'\,]_6\ [\,c'\,]_6$ (**6.3**)

$(c'\ ,\ out)$ (**6.4**)

$\boxed{\text{Rules in } \mathcal{R}_7}$

Rules to produce $2^{n\cdot m}$ copies of object c'' in the skin membrane at configuration $\mathcal{C}_{n^3+1}$:

$\{(c''_r\,,\, out\,;\, c''_{r-1}\,,\, in) \mid\ n\cdot m+1 \le r \le n^3\}$ (**7.1**)

$\{[\,c''_r\,]_7 \rightarrow [\,c''_{r-1}\,]_7\ [\,c''_{r-1}\,]_7 \mid 2 \le r \le n\cdot m\}$ (**7.2**)

$[\,c''_1\,]_7 \rightarrow [\,c''\,]_7\ [\,c''\,]_7;$ (**7.3**)

$(c''\ ,\ out)$ (**7.4**)

$\boxed{\text{Rules in } \mathcal{R}_8}$

Rules to produce $2^{n\cdot m}$ copies of object c''' in the skin membrane at configuration $\mathcal{C}_{n^3+1}$:

$\{(c'''_r\,,\, out\,;\, c'''_{r-1}\,,\, in) \mid n\cdot m+1 \le r \le n^3\}$ (**8.1**)

$\{[\,c'''_r\,]_8 \rightarrow [\,c'''_{r-1}\,]_8\ [\,c'''_{r-1}\,]_8 \mid 2 \le r \le n\cdot m\}$ (**8.2**)

$[\,c'''_1\,]_8 \rightarrow [\,c'''\,]_8\ [\,c'''\,]_8$ (**8.3**)

$(c'''\ ,\ out)$ (**8.4**)

$\boxed{\text{Rules in } \mathcal{R}_9}$

Rules to produce $2^{n\cdot m}$ copies of object c'''' in the skin membrane at configuration $\mathcal{C}_{n^3+1}$:

$\{(c''''_r\,,\, out\,;\, c''''_{r-1}\,,\, in) \mid n\cdot m+1 \le r \le n^3\}$ (**9.1**)

$\{[\,c''''_r\,]_9 \rightarrow [\,c''''_{r-1}\,]_9\ [\,c''''_{r-1}\,]_9 \mid 2 \le r \le n\cdot m\}$ (**9.2**)

$[\,c''''_1\,]_9 \rightarrow [\,c''''\,]_9\ [\,c''''\,]_9$ (**9.3**)

$(c''''\ ,\ out)$ (**9.4**)

$\boxed{\text{Rules in } \mathcal{R}_{10}}$

Rules to produce an affirmative answer:

$(\alpha_{n^3+6}\, c''''\,,\, in)\ ;\ (c''''\,\texttt{yes}\,,\, out)$ (**10.1**)

$\boxed{\text{Rules in } \mathcal{R}_{11}}$

Rules to control the negative answer of the computations by counters β_r:

$\{(\beta_r\, out\,;\, \beta_{r+1}\,,\, in) \mid\ 0 \le r \le n^3+6\}$ (**11.1**)

$(\beta_{n^3+7}\,\texttt{no}\,,\, out)$ (**11.2**)

$\boxed{\text{Rules in } \mathcal{R}_{a_{1,j}},\ 1 \le j \le n}$

Rules to produce 2^{n^3} copies of object a' in the skin membrane at configuration $\mathcal{C}_{n^3+1}$:

$\{[\,a'_r\,]_{a_{1,j}} \rightarrow [\,a'_{r-1}\,]_{a_{1,j}}\ [\,a'_{r-1}\,]_{a_{1,j}}\ \mid\ 2 \le r \le n^3\}$ (**a₁,ⱼ.1**)

$[\,a'_1\,]_{a_{1,j}} \rightarrow [\,a'\,]_{a_{1,j}}\ [\,a'\,]_{a_{1,j}}$ ($\mathbf{a_{1,j}.2}$)

$(a'\,,\, out)$ ($\mathbf{a_{1,j}.3}$)

$\boxed{\text{Rules in } \mathcal{R}_{a_{2,j}},\ 1 \le j \le n}$

Rules to produce 2^{n^3} copies of object a'' in the skin membrane at configuration $\mathcal{C}_{n^3+1}$:

$\{[\,a''_r\,]_{a_{2,j}} \rightarrow [\,a''_{r-1}\,]_{a_{2,j}}\ [\,a''_{r-1}\,]_{a_{2,j}}\ \mid\ 2 \le r \le n^3\}$ ($\mathbf{a_{2,j}.1}$)

$[\,a''_1\,]_{a_{2,j}} \rightarrow [\,a''\,]_{a_{2,j}}\ [\,a''\,]_{a_{2,j}}$ ($\mathbf{a_{2,j}.2}$)

$(a''\,,\, out)$ ($\mathbf{a_{2,j}.3}$)

$\boxed{\text{Rules in } \mathcal{R}_{e_{i,j,k}},\ 1 \le i,\, j,\, k \le n}$

Rules to produce 2^{n^3} copies of object $(i,\,j)''_k$ in the skin membrane at configuration $\mathcal{C}_{n^3+1}$:

$\{[(i,\,j)''_{k,r}]_{e_{i,j,k}} \rightarrow [(i,\,j)''_{k,r-1}]_{e_{i,j,k}}\, [(i,\,j)''_{k,r-1}]_{e_{i,j,k}} \mid 2 \le r \le n^3\}$ ($\mathbf{e_{i,j,k}.1}$)

$[\,(i,\,j)''_{k,1}\,]_{e_{i,j,k}} \rightarrow [\,(i,\,j)''_k\,]_{e_{i,j,k}}\ [\,(i,\,j)''_k\,]_{e_{i,j,k}}$ ($\mathbf{e_{i,j,k}.2}$)

$((i,\,j)''_k\,,\, out)$ ($\mathbf{e_{i,j,k}.3}$)

(5) The input membrane is the membrane labelled by 2 and the output region is the environment of the system (labelled by 0).

According with the previous description, for each $m, n \in \mathbb{N}$ the recognizer P system $\Pi(\langle m, n\rangle)$ is in the class **CDC**(2). In the next subsection, we will give an overview how each system $\Pi(\langle m, n\rangle)$ will process all directed graphs with n nodes and m arcs.

4.2. An overview of the computations

We consider the polynomial encoding (cod, s) from HAM-CYCLE to $\mathbf{\Pi}$ defined as follows: $cod(G) = \{(i, j)_k \mid (i, j) \in E, 1 \leq k \leq n\}$ and $s(G) = \langle m, n\rangle$, for each instance $G = (\{1, \dots, n\}, \{(i_1, j_1), \dots, (i_m, j_m)\})$. The expression $(i, j)_k$ in $cod(G)$ can be interpreted as follows: arc (i, j) is "placed" in "position k" in a potential path of G. According to this encoding, graph G will be processed by system $\Pi(s(G)) + cod(G)$. In what follows, we informally describe how system $\Pi(s(G)) + cod(G)$ works. The solution proposed implements the following stages:

- *Generation Stage*: All possible combinations of arcs from the input graph, including a code of their position in potential paths, are generated by using cell division in an adequate way.
- *Checking Stage*: It is checked whether or not the different combinations of arcs generated in the previous stage encode Hamiltonian cycles of the input graph.
- *Output Stage*: The system sends the right answer to the environment according to the results obtained in the previous stage.

Generation stage

At this stage, the system generates all the possible subsets of arcs of the graph (in fact, subsets of A'_G) which contain their potential positions in a path according to the notations introduced in Subsection 2.2. In this way, by applying rules of type **2.1** at configuration $\mathcal{C}_{n^3}$, there will be $2^{n \cdot m}$ membranes labelled by 2 such that each of them encodes a different combination of arcs from the input graph. For this, rules from $\mathcal{R}_2$ produce two new membranes for each object $(i, j)_k$: one containing its primed version, and other one not containing it. It produces that each membrane labelled by 2 will contain a different subset of A'_G, and therefore a different path if it is correct. Simultaneously, by applying rules of types 1, 2 and 3 from $\mathcal{R}_3$, $\mathcal{R}_4$, $\mathcal{R}_5$, $\mathcal{R}_6$, $\mathcal{R}_7$, $\mathcal{R}_8$ and $\mathcal{R}_9$, $2^{n \cdot m}$ copies of objects $b', b'', b''', c', c'', c'''$ and c'''' are produced in membranes labelled by $3, 4, 5, 6, 7, 8, 9$, respectively, and 2^{n^3} copies of objects a', a'' and $(i, j)''_k$ are produced in membranes labelled by $a_{1,j}, a_{2,j}$, and $e_{i,j,k}$, respectively. The generation stage takes n^3 steps.

Proposition 4.1. *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_q)$ be a computation of the system $\Pi(s(G)) + cod(G)$. Then at configuration $\mathcal{C}_{n^3}$, the following holds:*

1. *There are $2^{n \cdot m}$ membranes labelled by 2 such that each of them contains a different subset of $A'_G = \{(i, j)'_k \mid 1 \leq i, j, k \leq n, \ (i, j) \in E\}$ as well as object b, object c and n copies of object a.*
2. *There are $2^{n \cdot m}$ membranes labelled by 3 (respectively, by $4, 5, 6, 7, 8, 9$) such each of them only contains object b' (respectively, b'', b''', c', c'', c''' and c'''').*
3. *For each i, j, k $(1 \leq i, j, k \leq n)$ there are 2^{n^3} membranes labelled by $e_{i,j,k}$, each of them only containing object $(i, j)''_k$.*
4. *For each j $(1 \leq j \leq n)$ there are 2^{n^3} membranes labelled by $a_{1,j}$, each of them only containing object a', and there are 2^{n^3} membranes labelled by $a_{2,j}$, each of them only containing object a''.*
5. *There is a membrane labelled by 10 and a membrane labelled by 11 such that $\mathcal{C}_{n^3}(10) = \{\mathtt{yes}\}$, $\mathcal{C}_{n^3}(11) = \{\mathtt{no}, \beta_{\mathrm{n}^3}\}$.*
6. *The contents of the skin membrane, labelled by 1, is*
$$\mathcal{C}_{n^3}(1) = \{\alpha_{n^3}, \beta_0, \dots, \widehat{\beta_{n^3}}, \dots, \beta_{n^3+7}, b'_1, \dots b'_{n^3-1}, b''_1, \dots b''_{n^3-1}, b'''_1, \dots b'''_{n^3-1}, \\ c'_1, \dots c'_{n^3-1}, c''_1, \dots c''_{n^3-1}, c'''_1, \dots c'''_{n^3-1}, c''''_1, \dots c''''_{n^3-1}\}.$$

Checking stage

This stage takes 5 steps. At this stage, the system checks whether or not there exists a membrane labelled by 2 at configuration $\mathcal{C}_{n^3+5}$ which contains a subset of A''_G that encodes a Hamiltonian cycle of G.

At step $n^3 + 1$, the contents of membranes labelled by $3, 4, 5, 6, 7, 8, 9$, $a_{1,j}$ $(1 \leq j \leq n)$, $a_{2,j}$ $(1 \leq j \leq n)$ and $e_{i,j,k}$ $(1 \leq i, j, k \leq n)$ are sent to the skin membrane by applying rules 3.4, 4.4, 5.4, 6.4, 7.4, 8.4, 9.4, $a_{1,j}$.2, $a_{2,j}$.2, $e_{i,j,k}$.3. From this moment on, these membranes do not participate in the evolution of the configurations.

Proposition 4.2. *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_q)$ be a computation of the system $\Pi(s(G)) + cod(G)$. Then, omitting the empty membranes, at configuration in $\mathcal{C}_{n^3+1}$ we have the following:*

1. *There are $2^{n \cdot m}$ membranes labelled by 2 such that each of them contains a different subset of $A'_G = \{(i, j)'_k \mid 1 \leq i, j \leq m, 1 \leq k \leq n, \ (i, j) \in E\}$ as well as object b, object c and n copies of object a.*
2. *There is a membrane labelled by 10 and a membrane labelled by 11 such that $\mathcal{C}_{n^3+1}(10) = \{\mathtt{yes}\}$, $\mathcal{C}_{n^3+1}(11) = \{\mathtt{no}, \beta_{\mathrm{n}^3+1}\}$.*
3. *The skin membrane, labelled by 1, contains 2^{n^3} copies of each of the following objects: $(i, j)'_k$, $(i, j)''_k (1 \leq i, j, k \leq n)$, $a', a'', b', b'', b''', c', c'', c''', c''''$; also, it contains the following set of objects:*

$$\{\alpha_{n^3}, \beta_0, \ldots, \widehat{\beta_{n^3+1}}, \ldots, \beta_{n^3+7}, b'_1, \ldots b'_{n^3-1}, b''_1, \ldots b''_{n^3-1}, b'''_1, \ldots b'''_{n^3-1},$$
$$c'_1, \ldots c'_{n^3-1}, c''_1, \ldots c''_{n^3-1}, c'''_1, \ldots c'''_{n^3-1}, c''''_1, \ldots c''''_{n^3-1}\}.$$

At step n^3+2, objects a, b, c in membrane labelled by 2 at configuration $\mathcal{C}_{n^3+1}$ are substituted by objects a', b', c' from the skin membrane by applying rules **2.2**, **2.3**, and **2.4**. Simultaneously, by applying rules **2.6**, each subset of A'_G contained in a membrane labelled by 2 at configuration $\mathcal{C}_{n^3+1}$ produces the "corresponding" subset of A''_G. Besides, $\mathcal{C}_{n^3+2}(10) = \{\texttt{yes}\}$ and $\mathcal{C}_{n^3+2}(11) = \{\beta_{n^3+2}, \texttt{no}\}$.

At step n^3+3, by applying rules **2.3** and **2.4**, objects a', b', c' in membranes labelled by 2 at configuration $\mathcal{C}_{n^3+2}$ are substituted by objects a'', b'', c'' from the skin membrane, and simultaneously by applying rules of type **2.7** each subset contained in a membrane labelled by 2 at configuration $\mathcal{C}_{n^3+2}$ is transformed into a subset encoding all possible paths in the input graph. In this manner, according to Proposition 2.1, we have that the input graph (with n nodes and m arcs) has a Hamiltonian cycle if and only if at configuration $\mathcal{C}_{n^3+3}$, there exists some membrane labelled by 2 at configuration $\mathcal{C}_{n^3+3}$ such that the subset of A''_G contained in it has size equal to n. Besides, $\mathcal{C}_{n^3+3}(10) = \{\texttt{yes}\}$ and $\mathcal{C}_{n^3+3}(11) = \{\beta_{n^3+3}, \texttt{no}\}$.

At step n^3+4, by applying rules **2.3** and **2.4**, objects b'', c'' in membranes labelled by 2 are substituted by objects b''', c''' from the skin membrane, and simultaneously by applying rules **2.8** each object contained in the subset associated to each membrane labelled by 2 at configuration $\mathcal{C}_{n^3+3}$, is sent to the skin membrane cooperating with object a''. Then, the number of copies of object a'' appearing in a membrane labelled by 2 at configuration $\mathcal{C}_{n^3+4}$ is equal to $n-\gamma$, where γ is the size of the path in the input graph encoded by that membrane. Then, the input graph (with n nodes and m arcs) has a Hamiltonian cycle if and only if there exists a membrane labelled by 2 at configuration $\mathcal{C}_{n^3+4}$ such that it does not contain any object a''.

At step n^3+5, by applying rules of type **2.5**, objects a'' and b''' in membrane labelled by 2 at configuration $\mathcal{C}_{n^3+5}$ are sent to the skin membrane and, simultaneously, rule $(c''', out; c'''', in)$ produces an object c'''' in each membrane labelled by 2 at configuration $\mathcal{C}_{n^3+5}$. Hence, we can deduce the following proposition.

Proposition 4.3. *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_q)$ be a computation of the system $\Pi(s(G)) + cod(G)$. Then, the input graph has a Hamiltonian cycle if and only if there exists a membrane labelled by 2 at configuration $\mathcal{C}_{n^3+5}$ such that it contains an object b'''. Besides, $\mathcal{C}_{n^3+5}(10) = \{\texttt{yes}\}$ and $\mathcal{C}_{n^3+5}(11) = \{\beta_{n^3+5}, \texttt{no}\}$.*

Output stage

Finally, the output stage takes 3 steps. Only membranes labelled by 2 at configuration $\mathcal{C}_{n^3+5}$ which contain some object b''' (that is, membrane encoding a Hamiltonian cycle) can evolve and only rule $(c''', out; c'''', in) \in \mathcal{R}_2$ is applicable to that membrane. In this case, an object c'''' will appear in each membrane labelled by 2 at that configuration. Furthermore, if a membrane with label 2 at the mentioned configuration does not encode a Hamiltonian cycle of the input graph, then it contains objects b'', so the rule $(a''b''', out) \in \mathcal{R}_2$ will be applied. That is, the input graph has a Hamiltonian cycle if and only if some object c'''' appears in the skin membrane at configuration $\mathcal{C}_{n^3+6}$. Besides, $\mathcal{C}_{n^3+6}(10) = \{\texttt{yes}\}$ and $\mathcal{C}_{n^3+6}(11) = \{\beta_{n^3+6}, \texttt{no}\}$.

If the input graph has a Hamiltonian cycle, then only rules $(\alpha_{n^3+6}c'''', in) \in \mathcal{R}_{10}$ and $(\beta_{n^3+6}, out; \beta_{n^3+7}, in) \in \mathcal{R}_{11}$ are applicable to configuration $\mathcal{C}_{n^3+6}$. Otherwise, only rule $(\beta_{n^3+6} out; \beta_{n^3+7}, in)$ is applicable to that configuration. So, the answer to the problem is affirmative if and only if $\mathcal{C}_{n^3+7}(10) = \{\alpha_{n^3+6}c'''', \texttt{yes}\}$. Besides, in any case, $\mathcal{C}_{n^3+7}(11) = \{\beta_{n^3+7}, \texttt{no}\}$. Then, if there exists a Hamiltonian path then rules $(c''''\texttt{yes}, out) \in \mathcal{R}_{10}$ and $(\beta_{n^3+7}\texttt{no}, out) \in \mathcal{R}_{11}$ are applicable to configuration $\mathcal{C}_{n^3+7}$. Otherwise, only rule $(\beta_{n^3+7}\texttt{no}, out) \in \mathcal{R}_{11}$ is applicable to that configuration. Hence, the answer to the problem is affirmative if and only if the skin membrane at configuration $\mathcal{C}_{n^3+8}$ contains object $\texttt{yes}$ (together with objects $c'''', \beta_{n^3+7}, \texttt{no}$) but no object α_{n^3+6}. Otherwise, the skin membrane at configuration $\mathcal{C}_{n^3+8}$ contains objects $\beta_{n^3+7}, \texttt{no}, \alpha_{n^3+6}$ but no object $\texttt{yes}$.

At the last step, in case of an affirmative answer, rule $(\texttt{yes}, out)$ is applied to configuration $\mathcal{C}_{n^3+8}$ producing an object $\texttt{yes}$ in the environment and the computation halts. Otherwise, rule $(\texttt{no}\,\alpha_{n^3+6}, out)$ is applied to that configuration producing a negative answer.

Remark. When we provide a polynomial-time solution for an **NP**-hard problem by means of a family of recognizer membrane system, we are trading space for time, that is, an exponential number of membranes and/or objects are used along the computations. Let us notice that in the solution for the HAM-CYCLE problem, the maximum number of membranes used in any configuration is $(2n+n^3)\cdot 2^{n^3} + 8\cdot 2^{n\cdot m} + 3$, and the maximum number of objects used in any configuration is $(2n+n^3)\cdot 2^{n^3} + (n+10)\cdot 2^{n\cdot m} + 8n^3 + 3$.

4.3. Main result

Theorem 4.4. HAM-CYCLE $\in$ $\mathbf{PMC_{CDC(2)}}$.

Proof. The family of recognizer P systems designed in Section 4.1 verifies the following:

(a) Every system of the family $\mathbf{\Pi}$ is in the class $\mathbf{CDC}(2)$.
(b) The family $\mathbf{\Pi}$ is polynomially uniform by Turing machines. Indeed, for each $m, n \in \mathbb{N}$, the rules of $\Pi(\langle m, n\rangle)$ of the family are recursively defined from $m, n \in \mathbb{N}$, and the amount of resources needed to build $\Pi(\langle m, n\rangle)$ is of a polynomial order in n.
   - Size of the alphabet: $n^6 + 12n^3 + 29 \in \Theta(n^6)$;
   - Initial number of membranes: $n^3 + 2n + 11 \in \Theta(n^3)$;
   - Initial number of objects: $9n^3 + 3n + 13 \in \Theta(n^3)$;
   - Number of rules: $n^6 + 4n^5 + n^4 + 13n^3 + 2n + 30 \in \Theta(n^6)$;
   - Maximal length of a rule: $2 \in \Theta(1)$.

(c) The pair (cod, s) of polynomial-time computable functions defined in Subsection 4.2 has the following property: for each instance G of HAM-CYCLE, $s(G)$ is a natural number, $cod(G)$ is an input multiset of the system $\Pi(s(G))$, and for each $t \in \mathbb{N}$, $s^{-1}(t)$ is a finite set.
(d) The family $\mathbf{\Pi}$ is polynomially bounded, sound and complete with regard to (HAM-CYCLE, cod, s) (see Subsection 4.2).

Therefore, according to Definition 3.2, the family $\mathbf{\Pi}$ of P systems constructed in subsection 4.1 solves the HAM-CYCLE problem in polynomial-time with respect to the number of nodes. □

Corollary 4.5. $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathbf{CDC}(2)}$.

Proof. It suffices to notice that the HAM-CYCLE problem is an **NP**-complete problem, HAM-CYCLE$\in \mathbf{PMC}_{\mathbf{CDC}(2)}$, and the complexity class $\mathbf{PMC}_{\mathbf{CDC}(2)}$ is closed under polynomial-time reduction and under complement. □

5. Conclusions and open problems

This work should be considered as a contribution to the development of new mechanisms and tools in the framework of *Membrane Computing* to address the **P** versus **NP** problem.

In previous works, the computational efficiency of tissue-like P systems with symport/antiport rules and cell division has been studied in terms of cooperation among objects to apply a rule. It is worth pointing out that, in such framework, the use of cooperative rules (needing at least two objects to be triggered) is needed to solve **NP**-complete problems in uniform polynomial-time, assuming that $\mathbf{P} \neq \mathbf{NP}$ [29].

Cell-like P systems with symport/antiport rules and membrane division (only for elementary membranes) are a restrictive variant of such kind of tissue P systems. Recently, a uniform linear-time solution to the SAT problem has been provided by means of a family of this sort of cell-like P systems allowing communication rules involving at most three objects [12]. In this paper, this result has been improved by showing that *minimal cooperation* (two objects involved in the communication rules) is enough in order to solve efficiently computationally hard problems. Specifically, a uniform polynomial-time solution to the HAM-CYCLE problem by a family of P systems with membrane division which uses communication rules with length at most two, has been provided. Consequently, bearing in mind that $\mathbf{PMC}_{\mathbf{CDC}(1)} = \mathbf{P}$, a new *optimal tractability frontier* has been obtained in terms of the length of communication rules (total number of objects involved in them): in the framework of P systems with symport/antiport rules and membrane division (for elementary membranes), passing from 1 to 2 amounts to passing from non-efficiency to efficiency, assuming that $\mathbf{P} \neq \mathbf{NP}$.

By using membrane separation rules instead of membrane division rules in the framework of cell-like P systems with symport/antiport rules, a boundary of the efficiency is obtained passing from 2 to 3 in the length of communication rules [11]. Then, we have highlighted an interesting difference between *replication* of objects (abstracted by means of division rules) and *distribution* of objects (abstracted by means of separation rules) with respect to a frontier of the efficiency.

We conclude by proposing some open problems related to the role of the *direction* in communication rules (only symport or only antiport rules are allowed) in this kind of cell-like P systems from a complexity view.

(a) The solution given of the HAM-CYCLE problem can be adapted in such manner that only antiport rules are considered but then their length is at most three. So, $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathbf{CDA}(3)}$. What can be said about the computational complexity class $\mathbf{PMC}_{\mathbf{CDA}(2)}$?
(b) Concerning the efficiency of cell-like P systems with membrane division, what happens if the communication rules allowed can only be of the symport type?
(c) At the initial configuration of a P system with symport/antiport rules, the symbols of the distinguished alphabet $\mathcal{E}$ appear in the environment in an *arbitrary number of copies*. What about the efficiency of recognizer P systems from **CDA** and **CDS** when the alphabet of the environment is an empty set?

Acknowledgements

This work was partially supported by Project TIN2017-89842-P co-financed by *Ministerio de Economía, Industria y Competitividad* (MINECO) of Spain, through the *Agencia Estatal de Investigación* (AEI), and by *Fondo Europeo de Desarrollo Regional* (FEDER) of the European Union.

References

[1] T.H. Cormen, C.E. Leiserson, R.L. Rivest, An Introduction to Algorithms, The MIT Press, Cambridge, Massachusetts, 1994.
[2] M.R. Garey, D.S. Johnson, Computers and Intractability A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, 1979.
[3] R. Gutiérrez-Escudero, M.J. Pérez-Jiménez, M. Rius-Font, Characterizing tractability by tissue-like P systems, in: Membrane Computing, 10th International Workshop, WMC 2009, Curtea de Arges, Romania, August 24–27, 2009, in: Lecture Notes in Comput. Sci., vol. 5957, 2010, pp. 289–300, Revised Selected and Invited Papers.
[4] M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Núñez, F.J. Romero-Campero, On the efficiency of cell-like and tissue-like recognizing membrane systems, Int. J. Intell. Syst. 24 (7) (2009) 747–765.
[5] M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Núñez, F.J. Romero-Campero, On the power of dissolution in P systems with active membranes, in: R. Freund, et al. (Eds.), Membrane Computing, 6th International Workshop, WMC 2005, Vienna, Austria, July 18–21, 2006, in: Lecture Notes in Comput. Sci., vol. 3850, 2005, pp. 224–240, Revised Selected and Invited Papers.
[6] M. Ionescu, Gh. Păun, T. Yokomori, Spiking neural P systems, Fund. Inform. 71 (2–3) (2006) 279–308.
[7] L.F. Macías-Ramos, M.A. Martínez-del-Amor, M.J. Pérez-Jiménez, A. Riscos-Núñez, L. Valencia-Cabrera, The role of the direction in tissue P systems with cell separation, J. Autom. Lang. Comb. 19 (1–4) (2014) 185–199.
[8] L.F. Macías-Ramos, M.J. Pérez-Jiménez, A. Riscos-Núñez, M. Rius-Font, The efficiency of tissue P systems with cell separation relies on the environment, in: E. Csuhaj-Varjú, et al. (Eds.), Membrane Computing – 13th International Conference, CMC 2012 Budapest, Hungary, August 28–31, 2012, in: Lecture Notes in Comput. Sci., vol. 7762, 2013, pp. 243–256, Revised Selected Papers.
[9] L.F. Macías-Ramos, M.J. Pérez-Jiménez, A. Riscos-Núñez, L. Valencia-Cabrera, Membrane fission versus cell division: when membrane proliferation is not enough, Theoret. Comput. Sci. 608 (2015) 57–65.
[10] L.F. Macías-Ramos, B. Song, T. Song, L. Pan, M.J. Pérez-Jiménez, Limits on efficient computation in P systems with symport/antiport, in: C. Graciani, Gh. Păun, A. Riscos-Núñez, L. Valencia-Cabrera (Eds.), Proceedings of the Fifteenth Brainstorming Week on Membrane Computing, Sevilla, Spain, January 31, February 3, 2017, Fénix Editora, 2017, pp. 147–160.
[11] L.F. Macías-Ramos, B. Song, L. Valencia-Cabrera, L. Pan, M.J. Pérez-Jiménez, Membrane fission: a computational complexity perspective, Complexity 21 (6) (2016) 321–334.
[12] L.F. Macías-Ramos, L. Valencia-Cabrera, B. Song, T. Song, L. Pan, M.J. Pérez-Jiménez, P-Lingua based simulator for P systems with symport/antiport rules, Fund. Inform. 139 (2) (2015) 211–227.
[13] C. Martín-Vide, J. Pazos, Gh. Păun, A. Rodriguez-Paton, Tissue P systems, Theoret. Comput. Sci. 296 (2) (2003) 295–326.
[14] L. Pan, T.-O. Ishdorj, P systems with active membranes and separation rules, J. Univ. Comput. Sci. 10 (5) (2004) 630–649.
[15] L. Pan, M.J. Pérez-Jiménez, Computational complexity of tissue–like P systems, J. Complexity 26 (3) (2010) 296–315.
[16] A. Păun, Gh. Păun, The power of communication: P systems with symport/antiport, New Gener. Comput. 20 (3) (2002) 295–305.
[17] A. Păun, Gh. Păun, G. Rozenberg, Computing by communication in networks of membranes, Internat. J. Found. Comput. Sci. 13 (6) (2002) 779–798.
[18] Gh. Păun, Computing with membranes, J. Comput. Syst. Sci. 61 (1) (2000) 108–143.
[19] Gh. Păun, P systems with active membranes: attacking NP-complete problems, J. Autom. Lang. Comb. 6 (2001) 75–90.
[20] Gh. Păun, Membrane Computing. An Introduction, Springer-Verlag, Berlin, 2002.
[21] Gh. Păun, M.J. Pérez-Jiménez, A. Riscos-Núñez, Tissue P system with cell division, Int. J. Comput. Commun. Control III (3) (2008) 295–303.
[22] Gh. Păun, G. Rozenberg, A. Salomaa (Eds.), The Oxford Handbook of Membrane Computing, Oxford University Press, Oxford, 2010.
[23] I. Pérez-Hurtado, M.J. Pérez-Jiménez, A. Riscos-Núñez, M.A. Gutiérrez, M. Rius-Font, On a partial affirmative answer for a Păun's conjecture, Internat. J. Found. Comput. Sci. 22 (1) (2011) 55–64.
[24] M.J. Pérez-Jiménez, An approach to computational complexity in Membrane Computing, in: G. Mauri, et al. (Eds.), Membrane Computing, 5th International Workshop, WMC5, in: Lecture Notes in Comput. Sci., vol. 3365, 2005, pp. 85–109, Revised Selected and Invited Papers.
[25] M.J. Pérez-Jiménez, A. Riscos-Núñez, M. Rius-Font, F.J. Romero Campero, A polynomial alternative to unbounded environment for tissue P systems with cell division, Int. J. Comput. Math. 90 (4) (2013) 760–775.
[26] M.J. Pérez-Jiménez, A. Riscos-Núñez, M. Rius-Font, L. Valencia-Cabrera, The relevance of the environment on the efficiency of tissue P systems, in: A. Alhazov, et al. (Eds.), Membrane Computing – 14th International Conference, CMC 2013 Chisinau, Republic of Moldova, August 20–23, 2013, in: Lecture Notes in Comput. Sci., vol. 8340, 2014, pp. 308–321, Revised Selected Papers.
[27] M.J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini, Complexity classes in models of cellular computing with membranes, Nat. Comput. 2 (3) (2003) 265–285.
[28] M.J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini, A polynomial complexity class in P systems using membrane division, J. Autom. Lang. Comb. 11 (4) (2006) 423–434.
[29] A.E. Porreca, N. Murphy, M.J. Pérez-Jiménez, An optimal frontier of the efficiency of tissue P systems with cell division, in: M. García-Quismondo, et al. (Eds.), Tenth Brainstorming Week on Membrane Computing, Vol. II, Seville, Spain, January 30–February 3, 2012, Report RGNC 01/2012, Fénix Editora, 2012, pp. 141–166.
[30] P systems website, http://ppage.psystems.eu.