A Computational Complexity Theory in Membrane Computing

Mario J. Pérez–Jiménez

Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
marper@us.es

Summary. In this paper, a computational complexity theory within the framework of Membrane Computing is introduced. Polynomial complexity classes associated with different models of cell-like and tissue-like membrane systems are defined and the most relevant results obtained so far are presented. Many attractive characterizations of $\mathbf{P} \neq \mathbf{NP}$ conjecture within the framework of a bio-inspired and non-conventional computing model are deduced.

1 Introduction

The main objective of Computability Theory is to define the informal idea of mechanical/algorithmic problems resolution in a rigorous way. Each formal definition of the said concept provides a computing model. However, a basic question is to determine the class of all the problems that can be solved by a computing model when using the algorithms defined in it. In any computing model which captures the informal idea of algorithm, there are undecidible problems, that is, problems that cannot be solved by using the algorithms of the model.

Analyzing an algorithm which solves a problem consists of determining an upper bound for the minimal resource requirements with which the problem can be solved. The said upper bound will be a function of the size of the instance of the problem. One of the main goals of Computational Complexity Theory is to provide bounds on the amount of resources necessary for every mechanical procedure (algorithm) that solves a given problem.

Usually, complexity theory deals with *decision problems* which are problems that require a "*yes*" or "*no*" answer. A *decision problem*, X, is a pair (I_X, θ_X) such that I_X is a language over a finite alphabet (whose elements are called *instances*) and θ_X is a total boolean function (that is, a predicate) over I_X.

Many abstract problems are not decision problems. For example, in combinatorial optimization problems some value must be optimized (minimized or maximized). In order to deal with such problems, they can be transformed into roughly equivalent decision problems by supplying a target/threshold value for the quantity to be optimized, and then asking whether this value can be attained.

A natural correspondence between decision problems and languages can be established as follows. Given a decision problem $X = (I_X, \theta_X)$, its associated language is $L_X = \{w \in I_X : \ \theta_X(w) = 1\}$. Conversely, given a language L, over an alphabet Σ, its associated decision problem is $X_L = (I_{X_L}, \theta_{X_L})$, where $I_{X_L} = \Sigma^*$, and $\theta_{X_L} = \{(x, 1) : \ x \in L\} \cup \{(x, 0) : \ x \notin L\}$.

The solvability of decision problems is defined through the recognition of the languages associated with them. Let M be a Turing machine with a working alphabet Γ and L a language over Γ. Assume that the result of any halting computation of M is *yes* or *no*. If M is a *deterministic* device, then we say that M *recognizes* or *decides* L whenever, for any string u over Γ, if $u \in L$, then the answer of M on input u is *yes* (that is, M accepts u), and the answer is *no* otherwise (that is, M rejects u). If M is a *non-deterministic* device, then we say that M *recognizes* or *decides* L if for any string u over Γ, $u \in L$ if and only if there exists a computation of M with input u such that the answer is *yes*.

Throughout this paper, it is assumed that each abstract problem has an associated fixed *reasonable encoding scheme* that describes the instances of the problem by means of strings over a finite alphabet. We do not define *reasonable* in a formal way, however, following [8], instances should be encoded in a concise way, without irrelevant information, and where relevant numbers are represented in binary form (or any fixed base other than 1). It is possible to use multiple reasonable encoding schemes to represent instances, but it is proved that the input sizes differ at most by a polynomial. The *size* $|u|$ of an instance u is the length of the string associated with it, in some reasonable encoding scheme.

Membrane computing is a young branch of natural computing initiated by Gh. Păun at the end of 1998 [20]. Membrane systems are very flexible and versatile devices.

P systems take multisets as input, usually in a unary fashion. Hence, it is important to be careful when asserting that a problem is polynomial-time solvable by membrane systems. In this context, polynomial-time solutions to **NP**–complete problems in the framework of membrane computing can be considered as *pseudo-polynomial* time solutions in the classical sense (see [8] and [25] for details).

The paper is organized as follows. In the next section, basic concepts are introduced related to cell-like membrane systems that are necessary to define the solution of decision problems in polynomial time. In Section 3, limitations to basic transition P systems are described from the point of view of computational efficiency. Section 4 presents the most relevant results on P systems with active membranes both with and without polarization. Section 5 is devoted to the study of polarizationless tissue P systems with active membranes from the point of view of computational efficiency. In this Section, important results which provide borderlines between efficiency and non-efficiency are presented. The paper ends with the proposal of several open problems within the framework of computational complexity in Membrane Computing.

2 Cell–like Recognizer Membrane Systems

Membrane Computing is a young branch of Natural Computing providing distributed parallel computational devices called *membrane systems*, which are inspired in some basic biological features of living cells, as well as in the cooperation of cells in tissues, organs and organisms.

In this area there are basically two ways to consider computational devices: cell–like membrane systems (*P systems*) and tissue–like membrane systems (*tissue P systems*). The first one uses membranes arranged hierarchically, inspired from the structure of the cell, and the second one uses membranes placed in the nodes of a graph, inspired from the cell inter–communication in tissues.

In the last years several computing models using powerful tools from Nature have been developed (because of this, they are known as *bio-inspired* models) and several solutions in polynomial time to **NP**–complete problems have been presented, making use of non-determinism and/or of an exponential amount of space. This is the reason why a practical implementation of such models (in biological, electronic, or other media) could provide a significant advance in the resolution of computationally hard problems.

Definition 1. *A* P system *(without input) of degree $q \geq 1$ is a tuple of the form $\Pi = (\Gamma, H, \mu, \mathcal{M}_1, \dots, \mathcal{M}_q, R, i_{out})$, where:*

1. *Γ is a working alphabet of objects, and H is a finite set of labels;*
2. *μ is a membrane structure (a rooted tree) consisting of q membranes injectively labeled by elements of H;*
3. *$\mathcal{M}_1, \dots, \mathcal{M}_q$ are strings over Γ describing the initial multisets of objects placed in the q initial regions of μ;*
4. *R is a finite set of developmental rules;*
5. *$i_{out} \in H$ or $i_{out} = env$ indicates the output region: in the case $i_{out} \in H$, for a computation to be successful there must be exactly one membrane with label i_{out} present in the halting configuration; in the case $i_{out} = env$, i_{out} is usually omitted from the tuple.*

Many variants of P systems can be obtained depending on the kind of *developmental rules* and the semantics which are considered. The *length of a rule* is the number of symbols necessary to write it, both its left and right sides.

If h is the label of a membrane, then $f(h)$ denotes the label of the father of the membrane labeled by h. We assume the convention that the father of the skin membrane is the environment (env).

Definition 2. *A* P system with input membrane *is a tuple (Π, Σ, i_{in}), where: (a) Π is a P system; (b) Σ is an (input) alphabet strictly contained in Γ such* *that the initial multisets are over the alphabet $\Gamma \setminus \Sigma$; and (c) i_{in} is the label of a distinguished (input) membrane.*

The difference between P systems with and without input membrane is not related to their computations, but only to their initial configurations. A P system Π without input has a single *initial configuration* $(\mu, \mathcal{M}_1, \ldots, \mathcal{M}_q)$. A P system (Π, Σ, h_i) with input has many *initial configurations*: for each multiset $m \in \Sigma^*$, the *initial configuration* associated with m is $(\mu, \mathcal{M}_1, \ldots, \mathcal{M}_{h_i} \cup m, \ldots, \mathcal{M}_q)$.

In order to solve decision problems, we define *recognizer P system.*

Definition 3. *A* recognizer P system *is a P system such that: (a) the working alphabet contains two distinguished elements* yes *and* no*; (b) all computations halt; and (c) if $\mathcal{C}$ is a computation of the system, then either object* yes *or object* no *(but not both) must have been sent to the output region of the system, and only at the last step of the computation.*

For recognizer P systems, a computation $\mathcal{C}$ is said to be an *accepting computation* (respectively, *rejecting computation*) if the object *yes* (respectively, *no*) appears (only) in the output region associated with the corresponding halting configuration of $\mathcal{C}$.

For technical reasons all computations are required to halt, but this condition can often be removed without affecting computational efficiency.

Throughout this paper, $\mathcal{R}$ denotes an arbitrary class of recognizer P systems.

2.1 Uniform families of P systems

Many formal machine models (e.g. Turing machines or register machines) have an infinite number of memory locations. At the same time, P systems, or logic circuits, are computing devices of finite size and they have a finite description with a fixed amount of initial resources (number of membranes, objects, gates, etc.). For this reason, in order to solve a decision problem a (possibly infinite) family of P systems is considered.

The concept of solvability in the framework of P systems also takes into account the pre-computational process of (efficiently) constructing the family that provides the solution. In this paper, the terminology *uniform family* is used to denote that this construction is performed by a *single* computational machine.

In the case of P systems with input membrane, the term uniform family is consistent with the usual meaning for Boolean circuits: a family $\mathbf{\Pi} = \{\Pi(n) : n \in \mathbf{N}\}$ is uniform if there exists a deterministic Turing machine which constructs the system $\Pi(n)$ from $n \in \mathbf{N}$ (that is, which on input 1^n outputs $\Pi(n)$). In such a family, the P system $\Pi(n)$ will process all the instances of the problem with numerical parameters (reasonably) encoded by n – the common case is that $\Pi(n)$ processes all instances of size n. Note that this means that, for these families of P systems with input membrane, further pre–computational processes are needed in order to (efficiently) determine which P system (and from which input) deals with a given instance of the problem. The concept of *polynomial encoding* introduced below tries to capture this idea.

In the case of P systems without input membrane a new notion arises: a family $\mathbf{\Pi} = \{\Pi(w) : \ w \in I_X\}$ *associated with a decision problem* $X = (I_X, \theta_X)$ is *uniform* (some authors [15, 34, 37] use the term *semi-uniform* here) if there exists a deterministic Turing machine which constructs the system $\Pi(w)$ from the instance $w \in I_X$. In such a family, each P system usually processes only one instance, and the numerical parameters and syntactic specifications of the latter are part of the definition of the former.

It is important to point out that, in both cases, the family should be constructed in an efficient way. This requisite was first included within the term uniform family (introduced by Gh. Păun [21]), but nowadays it is preferred to use the term *polynomially uniform by Turing machines* to indicate a uniform (by a single Turing machine) and effective (in polynomial time) construction of the family.

Definition 4. *A family* $\mathbf{\Pi} = \{\Pi(w) : \ w \in I_X\}$ *(respectively,* $\mathbf{\Pi} = \{\Pi(n) : n \in \mathbf{N}\}$*) of recognizer membrane systems without input membrane (resp., with input membrane) is* polynomially uniform by Turing machines *if there exists a deterministic Turing machine working in polynomial time which constructs the system* $\Pi(w)$ *(resp.,* $\Pi(n)$*) from the instance* $w \in I_X$ *(resp., from* $n \in \mathbf{N}$*).*

2.2 Confluent P systems.

In order for recognizer P systems to capture the true algorithmic concept, a condition of *confluence* is imposed, in the sense that all possible successful computations must give the same answer. This contrasts with the standard notion of accepting computations for non-deterministic (classic) models.

Definition 5. *Let* $X = (I_X, \theta_X)$ *be a decision problem, and* $\mathbf{\Pi} = \{\Pi(w) : \ w \in I_X\}$ *be a family of recognizer P systems without input membrane.*

- $\mathbf{\Pi}$ *is said to be* sound with respect to X *if the following holds: for each instance of the problem,* $w \in I_X$*, if* there exists *an accepting computation of* $\Pi(w)$*, then* $\theta_X(w) = 1$.
- $\mathbf{\Pi}$ *is said to be* complete with respect to X *if the following holds: for each instance of the problem,* $w \in I_X$*, if* $\theta_X(w) = 1$*, then* every *computation of* $\Pi(w)$ *is an accepting computation.*

The concepts of soundness and completeness can be extended to families of recognizer P systems with input membrane in a natural way. However, an efficient process of selecting P systems from instances must be made precise.

Definition 6. *Let* $X = (I_X, \theta_X)$ *be a decision problem, and* $\mathbf{\Pi} = \{\Pi(n) : \ n \in \mathbf{N}\}$ *a family of recognizer P systems with input membrane. A* polynomial encoding *of* X *in* $\mathbf{\Pi}$ *is a pair* (cod, s) *of polynomial–time computable functions over* I_X *such that for each instance* $w \in I_X$*,* $s(w)$ *is a natural number (obtained by means of a reasonable encoding scheme) and* $cod(w)$ *is an input multiset of the system* $\Pi(s(w))$*.*

Polynomial encodings are stable under polynomial–time reductions [28].

Proposition 1. *Let X_1, X_2 be decision problems, r a polynomial–time reduction from X_1 to X_2, and (cod, s) a polynomial encoding from X_2 to $\mathbf{\Pi}$. Then, $(cod \circ r, s \circ r)$ is a polynomial encoding from X_1 to $\mathbf{\Pi}$.*

Next, the concepts of soundness and completeness are defined for families of recognizer P systems with input membrane.

Definition 7. *Let $X = (I_X, \theta_X)$ be a decision problem, $\mathbf{\Pi} = \{\mathit{\Pi}(n) : \ n \in \mathbf{N}\}$ a family of recognizer P systems with input membrane, and (cod, s) a polynomial encoding of X in $\mathbf{\Pi}$.*

- $\mathbf{\Pi}$ *is said to be* sound with respect to (X, cod, s) *if the following holds: for each instance of the problem, $w \in I_X$, if* there exists *an accepting computation of $\mathit{\Pi}(s(w))$ with input $cod(w)$, then $\theta_X(w) = 1$.*
- $\mathbf{\Pi}$ *is said to be* complete with respect to (X, cod, s) *if the following holds: for each instance of the problem, $w \in I_X$, if $\theta_X(w) = 1$, then* every *computation of $\mathit{\Pi}(s(w))$ with input $cod(w)$ is an accepting computation.*

Notice that if a family of recognizer P systems is sound and complete, then every P system of the family is confluent, in the sense previously mentioned.

2.3 Semi-Uniform Solutions versus Uniform Solutions

The first results showing that membrane systems could solve computationally hard problems in polynomial time were obtained using P systems without input membrane. In that context, a specific P system is associated with each instance of the problem. In other words, the syntax of the instance is part of the description of the associated P system. Thus this P system can be considered *special purpose.*

Definition 8. *A decision problem X is* solvable in polynomial time *by a family of recognizer P systems without input membrane $\mathbf{\Pi} = \{\mathit{\Pi}(w) : \ w \in I_X\}$, denoted by $X \in \mathbf{PMC}^*_{\mathcal{R}}$, if the following holds:*

- *The family $\mathbf{\Pi}$ is polynomially uniform by Turing machines.*
- *The family $\mathbf{\Pi}$ is polynomially bounded; that is, there exists a natural number $k \in \mathbf{N}$ such that for each instance $w \in I_X$, every computation of $\mathit{\Pi}(w)$ performs at most $|w|^k$ steps.*
- *The family $\mathbf{\Pi}$ is sound and complete with respect to X.*

The family $\mathbf{\Pi}$ is said to provide a *semi–uniform solution* to the problem X.

Next, recognizer P systems with input membrane are defined to solve problems in a *uniform* way in the following sense: all instances of a decision problem of the same *size* (via a given reasonable encoding scheme) are processed by the same system, to which an appropriate input is supplied.

Definition 9. *A decision problem $X = (I_X, \theta_X)$ is* solvable in polynomial time *by a family of recognizer P systems with input membrane $\mathbf{\Pi} = \{\Pi(n) : n \in \mathbf{N}\}$, denoted by $X \in \mathbf{PMC}_{\mathcal{R}}$, if the following holds:*

- *The family $\mathbf{\Pi}$ is polynomially uniform by Turing machines.*
- *There exists a polynomial encoding (cod, s) of X in $\mathbf{\Pi}$ such that:*
  - *The family $\mathbf{\Pi}$ is polynomially bounded with respect to (X, cod, s); that is, there exists a natural number $k \in \mathbf{N}$ such that for each instance $w \in I_X$, every computation of the system $\Pi(s(w))$ with input $cod(w)$ performs at most $|w|^k$ steps.*
  - *The family $\mathbf{\Pi}$ is sound and complete with respect to (X, cod, s).*

The family $\mathbf{\Pi}$ is said to provide a *uniform solution* to the problem X.

As a direct consequence of working with recognizer membrane systems, these complexity classes are closed under complement. Moreover, they are closed under polynomial–time reductions [28].

Obviously, every uniform solution of a decision problem provides a semi–uniform solution using the same amount of computational resources. That is, $\mathbf{PMC}_{\mathcal{R}} \subseteq \mathbf{PMC}^*_{\mathcal{R}}$, for any class $\mathcal{R}$ of recognizer P systems.

Remark: It is interesting to distinguish the concept of *polynomially uniform by Turing machines* from the concepts of *semi–uniform* and *uniform* solutions. The first concept is related with the resources required to construct the family of P systems solving a decision problem. The last two refer to the way in which the family processes the instances. In semi-uniform solutions, every instance is processed by a special purpose P system. While in uniform solutions, each P system processes all instances of a given size.

3 Efficiency of Basic Transition P Systems

In this section, the computational efficiency of P systems whose membrane structure does not increase is studied.

First of all, in order to formally define what means that a family of P systems simulates a Turing machine, we shall introduce for each Turing machine a decision problem associated with it.

Definition 10. *Let M be a Turing machine with input alphabet Σ_M. The* decision problem associated with M *is the problem $X_M = (I_M, \theta_M)$, where $I_M = \Sigma_M^*$, and for every $w \in \Sigma_M^*$, $\theta_M(w) = 1$ if and only if M accepts w.*

Obviously, the decision problem X_M is solvable by the Turing machine M.

Definition 11. *We say that a Turing machine M is simulated in polynomial time by a family of recognizer P systems from $\mathcal{R}$ if $X_M \in \mathbf{PMC}_{\mathcal{R}}$.*

A *basic transition* P system is a P system with only evolution, communication, and dissolution rules, which do not increase the size of the membrane structure. Let $\mathcal{T}$ denote the class of recognizer basic transition P systems.

M.A. Gutiérrez–Naranjo et al. [12] gave an efficient simulation of deterministic Turing machines by recognizer basic transition P systems.

Proposition 2. `(Sevilla theorem)` *Every deterministic Turing machine working in polynomial time can be simulated in polynomial time by a family of recognizer basic transition P systems with input membrane.*

They also proved that each confluent basic transition P system can be (efficiently) simulated by a deterministic Turing machine [12]. As a consequence, these P systems efficiently solve at most tractable problems.

Proposition 3. *If a decision problem is solvable in polynomial time by a family of recognizer basic transition P systems with input membrane, then there exists a deterministic Turing machine solving it in polynomial time.*

These results are also verified for recognizer basic transition P systems without input membrane. Therefore, the following holds.

Theorem 1. $\mathbf{P} = \mathbf{PMC}_{\mathcal{T}} = \mathbf{PMC}^*_{\mathcal{T}}$.

Thus, the ability of a P system in $\mathcal{T}$ to create exponential workspace (in terms of number of objects) in polynomial time (e.g. via evolution rules of the type $[\, a \rightarrow a^2 \,]_h$) is not enough to efficiently solve **NP**–complete problems (unless $\mathbf{P} = \mathbf{NP}$). Theorem 1 provides a tool to attack conjecture $\mathbf{P} = \mathbf{NP}$ in the framework of membrane computing.

Corollary 1. $\mathbf{P} \neq \mathbf{NP}$ *if and only if every, or at least one,* **NP***–complete problem is not in* $\mathbf{PMC}_{\mathcal{T}} = \mathbf{PMC}^*_{\mathcal{T}}$.

4 P Systems with Active Membranes

P systems with active membranes having associated electrical charges with membranes were first introduced by Gh. Păun [22]. Replication is one of the most important functions of a cell and, in ideal circumstances, a cell produces two identical copies by division (mitosis). Bearing in mind that the reactions which take place in a cell are related to membranes, rules for membrane division are considered.

Definition 12. *A P* system with active membranes *of degree* $q \geq 1$ *is a tuple* $\Pi = (\Gamma, H, \mu, \mathcal{M}_1, \dots, \mathcal{M}_q, R, i_{out})$*, where:*

1. Γ *is a working alphabet of objects, and* H *is a finite set of labels for membranes;*
2. μ *is a membrane structure (a rooted tree) consisting of* q *membranes injectively labeled by elements of* H*, and with electrical charges* $(+, -, 0)$ *associated with them;*

3. *$\mathcal{M}_1, \dots, \mathcal{M}_q$ are strings over Γ describing the initial multisets of objects placed in the q initial regions of μ;*
4. *R is a finite set of rules, of the following forms:*
   - *a)* $[\, a \rightarrow u \,]_h^{\alpha}$*, for* $h \in H, \alpha \in \{+,-,0\}$*,* $a \in \Gamma$*,* $u \in \Gamma^*$ (object evolution rules).
   - *b)* $a\,[\;\,]_h^{\alpha_1} \rightarrow [\, b \,]_h^{\alpha_2}$*, for* $h \in H$*,* $\alpha_1, \alpha_2 \in \{+,-,0\}$*,* $a, b \in \Gamma$ (send–in communication rules).
   - *c)* $[\, a \,]_h^{\alpha_1} \rightarrow [\;\,]_h^{\alpha_2}\, b$*, for* $h \in H$*,* $\alpha_1, \alpha_2 \in \{+,-,0\}$*,* $a, b \in \Gamma$ (send–out communication rules).
   - *d)* $[\, a \,]_h^{\alpha} \rightarrow b$*, for* $h \in H$*,* $\alpha \in \{+,-,0\}$*,* $a, b \in \Gamma$ (dissolution rules).
   - *e)* $[\, a \,]_h^{\alpha_1} \rightarrow [\, b \,]_h^{\alpha_2}\ [\, c \,]_h^{\alpha_3}$*, for* $h \in H$*,* $\alpha_1, \alpha_2, \alpha_3 \in \{+,-,0\}$*,* $a, b, c \in \Gamma$ (division rules for elementary membranes).
   - *f)* $[\,[\,]_{h_1}^{\alpha_1} \dots [\,]_{h_k}^{\alpha_1}\ [\,]_{h_{k+1}}^{\alpha_2} \dots [\,]_{h_n}^{\alpha_2}\,]_h^{\alpha} \rightarrow [\,[\,]_{h_1}^{\alpha_3} \dots [\,]_{h_k}^{\alpha_3}\,]_h^{\beta}\,[\,[\,]_{h_{k+1}}^{\alpha_4} \dots [\,]_{h_n}^{\alpha_4}\,]_h^{\gamma}$*, for* $k \geq 1$*,* $n > k$*,* $h, h_1, \dots, h_n \in H$*,* $\alpha, \beta, \gamma, \alpha_1, \dots, \alpha_4 \in \{+,-,0\}$ *and* $\{\alpha_1, \alpha_2\} = \{+,-\}$ (division rules for non–elementary membranes).
5. *$i_{out} \in H$ or $i_{out} = env$ indicates the output region.*

These rules are applied as usual (see [21] for details).

Note that these P systems have some important features: (a) they use three electrical charges; (b) the polarization of a membrane, but not the label, can be modified by the application of a rule; and (c) they do not use cooperation neither priorities.

In the framework of P systems without input membrane, C. Zandron et al. [39] proved that confluent recognizer P systems with active membranes making use of no membrane division rule, can be efficiently simulated by a deterministic Turing machine.

Proposition 4. (`Milano theorem`)
A deterministic P system with active membranes but without membrane division can be simulated by a deterministic Turing machine with a polynomial slowdown.

Let $\mathcal{NAM}$ be the class of recognizer P systems with active membranes which do not make use of division rules. As a consequence of the previous result, the following holds:

Corollary 2. $\mathbf{PMC}^*_{\mathcal{NAM}} \subseteq \mathbf{P}$.

A.E. Porreca [33] provides a simple proof of each tractable problem being able to be solved (in a semi–uniform way) by a family of recognizer P systems with active membranes (without polarizations) operating in exactly one step and using only send–out communication rules. That proof can be easily adapted to uniform solutions.

Proposition 5. $\mathbf{P} \subseteq \mathbf{PMC}_{\mathcal{NAM}}$.

Thus, we have a version of Theorem 1 for the class $\mathcal{NAM}$.

Theorem 2. $\mathbf{P} = \mathbf{PMC}_{\mathcal{NAM}} = \mathbf{PMC}^{*}_{\mathcal{NAM}}$.

The first efficient solutions to **NP**–complete problems by using P systems with active membranes were given in a *semi–uniform* way (where the P systems of the family depend on the syntactic structure of the instance) by S.N. Krishna et al. (`Hamiltonian Path`, `Vertex Cover` [13]), A. Obtulowicz (`SAT` [16]), A. Păun (`Hamiltonian Path` [19]), Gh. Păun (`SAT` [22, 23]), and C. Zandron et al. (`SAT`, `Undirected Hamiltonian Path` [39]).

Let $\mathcal{AM}(+n)$ (respectively, $\mathcal{AM}(-n)$) be the class of recognizer P systems with active membranes using division rules for elementary and non–elementary membranes (respectively, only for elementary membranes).

In the framework of $\mathcal{AM}(-n)$, efficient *uniform* solutions to weakly **NP**–complete problems (`Knapsack` [27], `Subset Sum` [26], `Partition` [10]), and strongly **NP**–complete problems (`SAT` [32], `Clique` [4], `Bin Packing` [30], `Common Algorithmic Problem` [29]) have been obtained.

Proposition 6. $\texttt{SAT} \in \mathbf{PMC}_{\mathcal{AM}(-n)}$.

Since $\mathbf{PMC}_{\mathcal{R}}$ is closed under complement and polynomial–time reductions, for any class $\mathcal{R}$ of recognizer P systems, the following result is obtained.

Proposition 7. $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{AM}(-n)}$.

In the framework of $\mathcal{AM}(+n)$, P. Sosík [37] gave an efficient *semi–uniform* solution to `QBF-SAT` (satisfiability of quantified propositional formulas), a well known **PSPACE**–complete problem [8]. Hence, the following is deduced.

Proposition 8. $\mathbf{PSPACE} \subseteq \mathbf{PMC}^{*}_{\mathcal{AM}(+n)}$.

This result has been extended by A. Alhazov et al. [5] showing that `QBF-SAT` can be solved in a linear time and in a *uniform* way by a family of recognizer P systems with active membranes (without using dissolution rules) and using division rules for elementary and non–elementary membranes.

Proposition 9. $\mathbf{PSPACE} \subseteq \mathbf{PMC}_{\mathcal{AM}(+n)}$.

A.E. Porreca et al. [34] described a (deterministic and efficient) algorithm simulating a single computation of any confluent recognizer P system with active membranes and without input. Such P systems can be simulated by a deterministic Turing machine working with exponential space, and spending a time of the order $O(2^{p(n)})$, for some polynomial $p(n)$. Thus,

Proposition 10. $\mathbf{PMC}^{*}_{\mathcal{AM}(+n)} \subseteq \mathbf{EXP}$.

Therefore, $\mathbf{PMC}_{\mathcal{AM}(+n)}$ and $\mathbf{PMC}^{*}_{\mathcal{AM}(+n)}$ are two membrane computing complexity classes between **PSPACE** and **EXP**.

Corollary 3. $\mathbf{PSPACE} \subseteq \mathbf{PMC}_{\mathcal{AM}(+n)} \subseteq \mathbf{PMC}^{*}_{\mathcal{AM}(+n)} \subseteq \mathbf{EXP}$.

P. Sosík et al. [36] have proven that the reverse inclusion of Proposition 8 holds as well. Nevertheless, the concept of *uniform family* of P systems considered in that paper is different from that of Definition 4, although maybe the proof can be adapted to fit into the framework presented in this paper. In this case the following would hold: $\mathbf{PSPACE} = \mathbf{PMC}^*_{\mathcal{AM}(+n)}$.

Previous results show that the usual framework of P systems with active membranes for solving decision problems is too powerful from the computational complexity point of view. Therefore, it would be interesting to investigate weaker models of P systems with active membranes able to characterize classical complexity classes below **NP** and providing borderlines between efficiency and non–efficiency.

Efficient (semi–uniform and/or uniform) solutions to computationally hard problems have been obtained within different apparently weaker variants of P systems with active membranes:

- P systems with separation rules instead of division rules, in two different cases: first one, using polarizations without changing membrane labels; and second one, without polarizations but allowing change of membrane labels (`SAT`, uniform solution [18]).
- P systems using division for elementary membranes, without changing membrane labels, without polarizations, but using bi–stable catalysts (`SAT`, uniform solution [31]).
- P systems using division for elementary membranes, without label changing, but using only two electrical charges (`SAT`, uniform solution [2], `Subset Sum`, uniform solution [35]).
- P systems without polarizations, without label changing, without division, but using three types of membrane rules: separation, merging, and release (`SAT`, semi–uniform solution [17]).
- P systems without dissolution nor polarizations, but allowing to change the labels of membranes in division rules (`SAT`, uniform solution [3]).
- P systems without dissolution nor polarizations, but allowing to change the labels of membranes in send–out rules (`SAT`, uniform solution [3]).
- P systems without polarizations, but using division for elementary and non–elementary membranes (`SAT`, semi–uniform solution [3]).

4.1 Polarizationless P systems with active membranes

Next, several classes of recognizer P systems with active membranes without electrical charges and with different kinds of membrane division rules are studied from a computational complexity point of view.

Definition 13. *A* polarizationless P system with active membranes *of degree* $q \geq 1$ *is a tuple* $\Pi = (\Gamma, H, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, R, i_{out})$*, where:*

1. *Γ is a working alphabet of objects, and H is a finite set of labels for membranes;*

2. *μ is a membrane structure (a rooted tree) consisting of q membranes injectively labeled by elements of H;*
3. *$\mathcal{M}_1, \dots, \mathcal{M}_q$ are strings over Γ describing the multisets of objects placed in the q initial regions of μ;*
4. *R is a finite set of developmental rules, of the following forms:*
   (a) $[\, a \to u \,]_h$*, for* $h \in H$*,* $a \in \Gamma$*,* $u \in \Gamma^*$ (object evolution rules).
   (b) $a\,[\ ]_h \to [\, b \,]_h$*, for* $h \in H$*,* $a, b \in \Gamma$ (send–in communication rules).
   (c) $[\, a \,]_h \to [\ ]_h\, b$*, for* $h \in H$*,* $a, b \in \Gamma$ (send–out communication rules).
   (d) $[\, a \,]_h \to b$*, for* $h \in H$*,* $a, b \in \Gamma$ (dissolution rules).
   (e) $[\, a \,]_h \to [\, b \,]_h\ [\, c \,]_h$*, for* $h \in H$*,* $a, b, c \in \Gamma$ (division rules for elementary *or weak* division rules for non-elementary membranes).
   (f) $[\,[\,]_{h_1} \dots [\,]_{h_k}\, [\,]_{h_{k+1}} \dots [\,]_{h_n}\,]_h \to [\,[\,]_{h_1} \dots [\,]_{h_k}\,]_h\ [\,[\,]_{h_{k+1}} \dots [\,]_{h_n}\,]_h$*, where* $k \geq 1$*,* $n > k$*,* $h, h_1, \dots, h_n \in H$ (*strong* division rules for non-elementary membranes).
5. *$i_{out} \in H$ or $i_{out} = env$ indicates the output region.*

These rules are applied according to usual principles of polarizationless P systems (see [11] for details).

Notice that in this polarizationless framework there is no cooperation, priority, nor changes of the labels of membranes. Besides, throughout this paper, rules of type (f) are used only for $k = 1, n = 2$, that is, rules of the form $(f)\ \ [\,[\ ]_{h_1}[\ ]_{h_2}\,]_h \to [\,[\ ]_{h_1}\,]_h\ [\,[\ ]_{h_2}\,]_h$. They can also be restricted to the case where they are controlled by the presence of a specific membrane, that is, rules of the form $(g)\ \ [\,[\ ]_{h_1}[\ ]_{h_2}[\ ]_p\,]_h \to [\,[\ ]_{h_1}[\ ]_p\,]_h\ [\,[\ ]_{h_2}[\ ]_p\,]_h$.

The class of recognizer polarizationless P systems with active membranes (resp., which do not make use of division rules) is denoted by $\mathcal{AM}^0$ (resp., $\mathcal{NAM}^0$), and $\mathcal{AM}^0(\alpha, \beta, \gamma, \delta)$, where $\alpha \in \{-d, +d\}$, $\beta \in D = \{-n, +nw, +ns, +nsw, +nsr\}$, $\gamma \in \{-e, +e\}$, and $\delta \in \{-c, +c\}$, denotes the class of all recognizer P systems with polarizationless active membranes such that:

(a) if $\alpha = +d$ (resp., $\alpha = -d$) then dissolution rules are permitted (resp., forbidden);
(b) if $\beta = +nw$ or $+ns$ (resp., $\beta = +nsw$) then division rules for elementary and non–elementary membranes, weak or strong (resp., weak and strong) are permitted; if $\beta = +nsr$ then division rules of the types (e), (f) and (g) are permitted; if $\beta = -n$ then only division rules for elementary membranes are permitted.
(c) if $\gamma = +e$ (resp., $\gamma = -e$) then evolution rules are permitted (resp., forbidden);
(d) if $\delta = +c$ (resp., $\delta = -c$) then communication rules are permitted (resp., forbidden).

Proposition 5 can be adapted to polarizationless P systems with active membranes which do not make use of division nor evolution rules, providing a lower bound about their efficiency.

Proposition 11. $\mathbf{P} \subseteq \mathbf{PMC}_{\mathcal{NAM}^0(-d,-e,+c)}$.

4.2 A conjecture of Păun

At the beginning of 2005, Gh. Păun (problem **F** from [24]) wrote:

My favorite question (related to complexity aspects in P systems with active membranes and with electrical charges) is that about the number of polarizations. Can the polarizations be completely avoided? The feeling is that this is not possible – and such a result would be rather sound: passing from no polarization to two polarizations amounts to passing from non–efficiency to efficiency.

This so–called Păun's conjecture can be formally formulated in terms of membrane computing complexity classes as follows:

$$\mathbf{P} = \mathbf{PMC}^{[*]}_{\mathcal{AM}^0(+d,-n,+e,+c)}$$

where the notation $\mathbf{PMC}^{[*]}_{\mathcal{R}}$ indicates that the result holds for both $\mathbf{PMC}_{\mathcal{R}}$ and $\mathbf{PMC}^{*}_{\mathcal{R}}$.

Let Π be a recognizer polarizationless P system with active membranes which do not make use of dissolution rules. A directed graph can be associated with Π verifying the following property: every accepting computation of Π is characterized by the existence of a path in the graph between two specific nodes.

Each rule of Π can be considered as a *dependency relation* between the object triggering the rule and the object(s) produced by its application. We can consider a general pattern for rules of types $(a), (b), (c), (e)$ in the form $(a, h) \rightarrow (a_1, h')(a_2, h') \dots (a_s, h')$, where the rules of type (a) correspond to the case $h = h'$, the rules of type (b) correspond to the case $h = f(h')$ and $s = 1$, the rules of type (c) correspond to the case $h' = f(h)$ and $s = 1$, and the rules of type (e) correspond to the case $h = h'$ and $s = 2$. A formal definition of the *dependency graph* associated with a P system can be found in [11].

Note that a P system can dynamically evolve according to its rules, but the dependency graph associated with it is static. Furthermore, rules of the kind (f) and (g) do not provide any node nor arc to the dependency graph.

Let Δ_Π be the set of all pairs $(a, h) \in \Gamma \times H$ such that there exists a path (within the dependency graph) from (a, h) to $(\texttt{yes}, env)$ – the environment is considered to be the output region, although the results obtained are also valid for any output membrane.

In [11] the following results are shown.

Proposition 12. *Let Π be a recognizer polarizationless P systems with active membranes not using dissolution rules, and where every kind of division rules is permitted. Then,*

- *There exists a Turing machine that constructs the dependency graph associated with Π in a time bounded by a polynomial function depending on the total number of rules and the maximum length of the rules.*

- *There exists a Turing machine that constructs the set $\varDelta_{\varPi}$ in a time bounded by a polynomial function depending on the total number of rules and the maximum length of the rules.*

Given a family $\mathbf{\Pi} = \{\varPi(n) : \ n \in \mathbf{N}\}$ of recognizer P systems solving a decision problem in a uniform way (with (cod, s) being the associated polynomial encoding), the acceptance of a given instance of the problem, w, can be characterized by using the set $\varDelta_{\varPi(s(w))}$ associated with $\varPi(s(w))$.

Let $\overline{\mathcal{M}_j} = \{(a, j) : \ a \in \mathcal{M}_j\}$, for $1 \leq j \leq q$ and $\overline{m} = \{(a, h_i) : \ a \in m\}$, for each input multiset m over $\varSigma$ (recall that h_i is the label of the input membrane). Then, the following holds [11]:

Proposition 13. *Let $X = (I_X, \theta_X)$ be a decision problem, and $\mathbf{\Pi} = \{\varPi(n) : \ n \in \mathbf{N}\}$ a family of recognizer polarizationless P systems and not using dissolution rules solving X in a uniform way. Let (cod, s) be a polynomial encoding associated with that solution. Then, for each instance w of the problem X the following statements are equivalent:*

(a) $\theta_X(w) = 1$ *(that is, the answer to the problem is* `yes` *for w).*

(b) $\varDelta_{\varPi(s(w))} \cap \big(\, \overline{cod(w)} \cup \bigcup_{j=1}^{q} \overline{\mathcal{M}_j}\,\big) \neq \emptyset$*, where $\mathcal{M}_1, \dots, \mathcal{M}_q$ are the initial multisets of $\varPi(s(w))$.*

A similar result holds for semi–uniform solutions [11] and the following theorem can be deduced.

Theorem 3. $\mathbf{P} = \mathbf{PMC}^{[*]}_{\mathcal{AM}^0\,(-d,\beta,+e,+c)}$*, where $\beta \in D$.*

Thus, polarizationless P systems with active membranes which do not make use of dissolution rules are non–efficient in the sense that their cannot solve **NP**–complete problems in polynomial time (unless **P**=**NP**).

Let us now consider polarizationless P systems with active membranes making use of dissolution rules. Will it be possible to solve **NP**–complete problems in that framework?

N. Murphy et al. [15] gave a negative answer in the case that division rules are used only for elementary membranes and being *symmetric*, in the following sense $[\,a\,]_h \rightarrow [\,b\,]_h[\,b\,]_h$.

Theorem 4. $\mathbf{P} = \mathbf{PMC}^{[*]}_{\mathcal{AM}^0\,(+d,-n(sym),+e,+c)}$.

D. Woods et al. [38] have recently provide a **P** upper bound on polarizationless P systems with dissolution and division only for elementary membranes, without evolution and communication rules, where at the initial timestep, the depth of membrane nesting is equal to the total number of membranes.

Theorem 5. *If $\mathcal{D}$ is the class of systems in $\mathcal{AM}^0\,(+d,-n,-e,-c)$, having an initial membrane structure that is a single (linear) path, then $\mathbf{P} = \mathbf{PMC}_{\mathcal{D}}^{[*]}$.*

Several authors [3, 11] gave a positive answer when division for non–elementary membranes, in the strong sense, is permitted. The mentioned papers provide semi–uniform solutions in a linear time to `SAT` and `Subset Sum`, respectively. Thus, we have the following result:

Proposition 14. $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}^{*}_{\mathcal{AM}^0\,(+d,+ns,+e,+c)}$.

As a consequence of Theorems 3 and 14, a *partial negative* answer to Păun's conjecture is given: assuming that $\mathbf{P} \neq \mathbf{NP}$ and making use of dissolution rules and division rules for elementary and non–elementary membranes, computationally hard problems can be efficiently solved avoiding polarizations. The answer is partial because efficient solvability of **NP**–complete problems by polarizationless P systems with active membranes making use of dissolution rules and division *only* for elementary membranes is unknown.

The result of Theorem 14 was improved by A. Alhazov et al. [1] giving a family of recognizer polarizationless P systems with active membranes using dissolution rules and division for elementary and (strong) non–elementary membranes solving `QBF-SAT` in a *uniform* way and in a linear time. Then,

Proposition 15. $\mathbf{PSPACE} \subseteq \mathbf{PMC}_{\mathcal{AM}^0\,(+d,+ns,+e,+c)}$.

Next, we present some results about the efficiency of polarizationless P systems with active membranes when evolution rules and/or communication rules are forbidden.

First, one can adapt a solution given in [3] to provide a semi-uniform solution to `SAT` in a linear time by a family of recognizer polarizationless P systems with active membranes by using evolution, dissolution and division rules for elementary and non–elementary membranes (both in the strong and weak versions), and avoiding communication rules. That is, we have the following:

Proposition 16. $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}^{*}_{\mathcal{AM}^0\,(+d,\beta,+e,-c)}$, *where* $\beta \in \{+nw, +ns\}$.

Evolution and communication rules can be avoided without loss of efficiency. Indeed, in [40] a semi–uniform solution to `3-SAT` in a linear time by a family of polarizationless recognizer P systems with active membranes by using only dissolution rules and division rules for elementary and non–elementary membranes of the types (e) and (f), is presented. Thus, the following holds:

Proposition 17. $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}^{*}_{\mathcal{AM}^0\,(+d,+nsw,-e,-c)}$.

Moreover, Proposition 17 can be extended when non–elementary membrane division controlled by the presence of a membrane is allowed. In [14] it was presented a semi–uniform solution to `QBF-3-SAT` in a linear time by a family of polarizationless recognizer P systems with active membranes by using only dissolution rules and division rules of the types (e), (f) and (g). Thus, the following holds:

Proposition 18. $\mathbf{PSPACE} \subseteq \mathbf{PMC}^*_{\mathcal{AM}^0\,(+d,+nsr,-e,-c)}$.

Figure 1 graphically summarize the results known related with complexity classes associated with polarizationless P systems with active membranes making use of dissolution rules. In the picture, $-u$ (resp. $+u$) means semi–uniform (resp. uniform) solutions, $-n$ (resp. $+ns$ or $+nsw$)) means using division only for elementary membranes (resp. division for elementary and non–elementary membranes in the *strong* version or *strong* and *weak* version), $-n(sym)$ means using division only for elementary membranes and being *symmetric*, $-ev$ (resp. $+ev$) means that evolution rules are forbidden (resp. permitted), and $-comm$ (resp. $+comm$) means that communication rules are forbidden (resp. permitted). A standard class inside (respectively, over) a dark node means that the corresponding membrane computing class is equal (resp., is a lower bound) to the standard class.

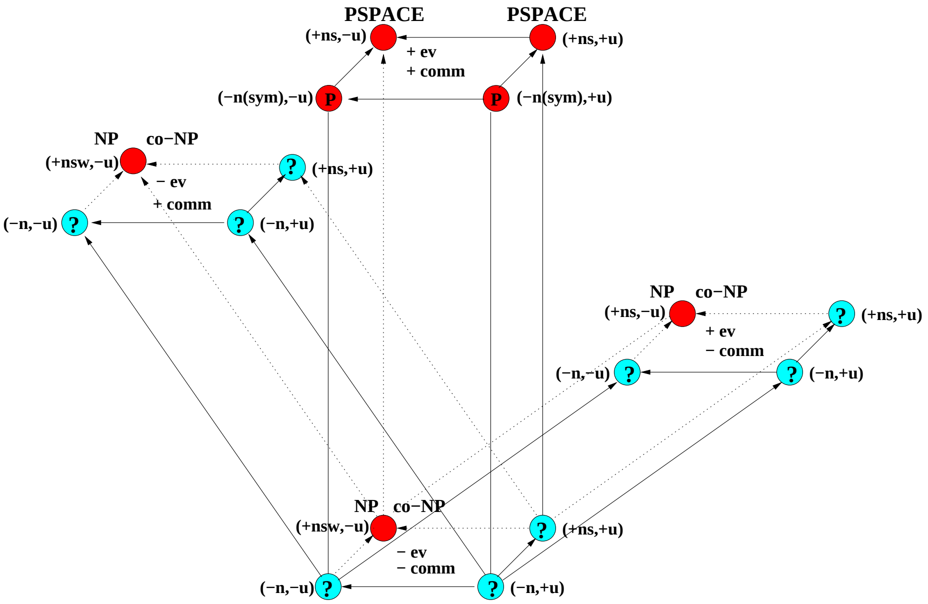

Fig. 1. Polarizationless active membranes *using dissolution rules*

5 Tissue–like Recognizer P systems with cell division

In this section, we consider computational devices inspired in cell inter–communication in tissues and we add the ingredient of cell division rules as we did to polarizationless P systems with active membranes (and with input membrane).

Definition 14. *A polarizationless tissue–like membrane system (tissue P system, for short) with cell division of degree $q \geq 1$ is a tuple*

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

where:

1. *Γ is the working alphabet containing two distinguished objects* `yes` *and* `no`*;*
2. *Σ is an (input) alphabet strictly contained in Γ.*
3. *$\Omega \subseteq \Gamma \setminus \Sigma$ is a finite alphabet, describing the set of objects located in the environment in an arbitrary number of copies each;*
4. *$\mathcal{M}_1, \dots, \mathcal{M}_q$ are multisets over $\Gamma - \Sigma$, describing the objects placed in the cells of the system (we suppose that at least one copy of* `yes` *and* `no` *is in some of these multisets);*
5. *R is a finite set of developmental rules, of the following forms:*
   a) $(i, u/v, j)$, for $i, j \in \{0, 1, 2, \dots, p\}, i \neq j$, and $u, v \in \Gamma^*$; $1, 2, \dots, p$ identify the cells of the system, 0 is the environment: When applying a rule $(i, u/v, j)$, the objects of the multiset represented by u are sent from region i to region j and the objects of the multiset v are sent from region j to region i simultaneously;
   b) $[\,a\,]_i \rightarrow [\,b\,]_i [\,c\,]_i$, where $i \in \{1, 2, \dots, p\}$ and $a, b, c \in \Gamma$: Under the influence of object a, the cell labeled by i is divided in two cells with the same label; object a is replaced by b in the first copy, object a is replaced by c in the second copy; all the other objects are replicated and copies of them are placed in the two new cells.
6. *$i_{in} \in \{1, \dots, q\}$ is the input cell, and $i_{out} \in \{0, 1, \dots, q\}$ is the output cell.*

Let m be a multiset over Σ. The *initial configuration of Π with input m* is tuple $(\mathcal{M}_1, \dots, \mathcal{M}_{i_{in}} \cup m, \dots, \mathcal{M}_p)$.

The rules of a tissue–like membrane system as the one above are used in a nondeterministic maximally parallel way as customary in membrane computing. At each step, we apply a set of rules which is maximal (no further rule can be added), with the following important restriction: if a cell is divided, then the division rule is the only one which is applied for that cell at that step, and so its objects do not participate in any communication rule.

All computations start from an initial configuration and proceed as stated above; only halting computations give a result, which is encoded by the number of objects in the output cell i_{out} in the last configuration. From now on, we will consider that the output is collected in the environment (that is, $i_{out} = 0$, and thus, we will omit i_{out} in the definition of tissue P systems). In this way, if Π is a tissue P system and $\mathcal{C} = \{C_i\}_{i<r}$ is a halting computation of Π, then the answer of the computation $\mathcal{C}$ is

$$Output(\mathcal{C}) = \Psi_{\Gamma \setminus \Omega}(M_{r-1,0})$$

where Ψ is the Parikh function, and $M_{r-1,0}$ is the multiset over $\Gamma \setminus \Omega$ associated with the environment at the halting configuration C_{r-1}.

Definition 15. *A polarizationless tissue–like membrane system with cell division is said to be a* recognizer *system if: (a) the working alphabet contains two distinguished elements* yes *and* no*; (b) all computations halt; and (c) if $\mathcal{C}$ is a computation of the system, then either object* yes *or object* no *(but not both) must have been sent to the output region of the system, and only at the last step of the computation.*

Given a recognizer tissue P system with cell division, and a computation $\mathcal{C} = \{C_i\}_{i<r}$ of Π ($r \in \mathbf{N}$), we define the result of $\mathcal{C}$ as follows:

$$Output(\mathcal{C}) = \begin{cases} \texttt{yes}, \text{ if } \Psi_{\{\texttt{yes},\texttt{no}\}}(M_{r-1,0}) = (1,0) \\ \qquad \wedge\, \Psi_{\{\texttt{yes},\texttt{no}\}}(M_{k,0}) \quad = (0,0) \text{ for } k = 0,\dots,r-2 \\ \texttt{no}, \;\; \text{if } \Psi_{\{\texttt{yes},\texttt{no}\}}(M_{r-1,0}) = (0,1) \\ \qquad \wedge\, \Psi_{\{\texttt{yes},\texttt{no}\}}(M_{k,0}) \quad = (0,0) \text{ for } k = 0,\dots,r-2 \end{cases}$$

That is, $\mathcal{C}$ is an accepting computation (respectively, rejecting computation) if the object `yes` (respectively, `no`) appears (only) in the environment associated with the halting configuration.

We denote by $\mathcal{TDC}$ (respectively, $\mathcal{TDC}(k)$) the class of recognizer tissue–like membrane systems with cell division (by using communication rules whose length is, at most, k). We also denote by $\mathcal{TC}$ the class of recognizer tissue–like recognizer membrane systems without cell division.

The concepts of polynomially uniform by Turing machines, polynomial encoding, polynomially bounded, soundness and completeness introduced at definitions 4, 6, 7 and 9 can be naturally generalized to the framework of recognizer tissue P systems. This allows us to define the concept of uniform solvability in polynomial time by using systems in $\mathcal{TDC}$.

Definition 16. *We say that a decision problem $X = (I_X, \theta_X)$ is solvable in polynomial time by a family $\mathbf{\Pi} = \{\Pi(n) : n \in \mathbb{N}\}$ of recognizer tissue P systems with cell division if the following holds:*

- *The family $\mathbf{\Pi}$ is* polynomially uniform *by Turing machines, that is, there exists a deterministic Turing machine which constructs the system $\Pi(n)$ from $n \in \mathbb{N}$ in polynomial time with respect to n.*
- *There exists a pair (cod, s) of polynomial-time computable functions over I_X (called a polynomial encoding of I_X in $\mathbf{\Pi}$) 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 $\Pi(s(u))$.*
  - *The family $\mathbf{\Pi}$ is* polynomially bounded *with regard to (X, cod, s); that is, there exists a polynomial function p, such that for each $u \in I_X$ every computation of $\Pi(s(u))$ with input $cod(u)$ is halting and, moreover, it performs at most $p(|u|)$ steps.*
  - *The family $\mathbf{\Pi}$ is* sound *with regard to (X, cod, s); that is, for each $u \in I_X$, if there exists an accepting computation of $\Pi(s(u))$ with input $cod(u)$, then $\theta_X(u) = 1$.*

– *The family* $\mathbf{\Pi}$ *is* complete *with regard to* (X, cod, s)*; that is, for each* $u \in I_X$*, if* $\theta_X(u) = 1$*, then every computation of* $\Pi(s(u))$ *with input* $cod(u)$ *is an accepting one.*

From the soundness and completeness conditions above we deduce that every P system $\Pi(n)$ is *confluent*, in the following sense: every computation of a system with the *same* input multiset must always give the *same* answer.

We denote by $\mathbf{PMC}_{\mathcal{R}}$ the set of all decision problems which can be solved by means of recognizer tissue P systems of $\mathcal{R}$ in polynomial time. This class is closed under complement and polynomial–time reductions (see [28] for a similar result for cell-like P systems).

In [7] a polynomial time solution of the Vertex Cover problem was given by using a family of recognizer tissue P systems with cell division and communication rules of length at most 3. Then

Proposition 19. $\mathbf{NP} \cup \textbf{co-NP} \subseteq \mathbf{PMC}_{TDC(3)}$.

5.1 Allowing communication rules of length at most 1

For recognizer tissue P systems with cell division and communication rules with length at most 1, it can be generalized the concept of dependency graph in a natural way.

We can consider a general pattern $(a, i) \rightarrow (b_1, j) \dots (b_s, j)$ where $i, j \in \{0, 1, 2, \dots, q\}, i \neq j$, and $a, b \in \Gamma$. Communication rules correspond to the case $s = 1$ and $b_1 = a$, and division rules correspond to the case $s = 2$ and $j = i \neq 0$. The above pattern can be interpreted as follows: from the object a in the cell (or in the environment) labeled by i we can *reach* objects $b_1, \dots, b_s$ in the cell (or in the environment) labeled by j.

By using the concept of dependency graph associated with P systems with cell division and communication rules with length at most 1, it has been proved that this kind of tissue P systems can only efficiently solve tractable problems (see [9], for details).

Theorem 6. $\mathbf{P} = \boldsymbol{PMC}_{TDC(1)}$

From Proposition 19 and Theorem 6, we deduce that in the framework of recognizer tissue P systems with cell division the length of the communication rules provides a borderline between efficiency and non-efficiency. Specifically, a frontier is obtained when we pass from length 1 to length 3.

6 Efficiency of Tissue P Systems without cell division

A family of recognizer tissue P systems with symport/antiport rules which solves a decision problem can be efficiently simulated by a family of basic recognizer P systems solving the same problem. This simulation allows us to transfer the result about the limitations in computational power, from the model of basic cell–like P systems to this kind of tissue–like P systems.

Definition 17. *Let Π and Π' be recognizer cellular systems (cell–like and/or tissue–like). We say that Π'* efficiently simulates *Π if the following holds:*

- *Π' can be constructed from Π by a deterministic Turing machine working in polynomial time.*
- *There exists a bijective function, f, from the set $\mathbf{Comp}(\Pi)$ of computations of Π onto the set $\mathbf{Comp}(\Pi')$ of computations of Π' such that:*
  - *A computation $\mathcal{C} \in \mathbf{Comp}(\Pi)$ is an accepting computation if and only if $f(\mathcal{C}) \in \mathbf{Comp}(\Pi')$ is an accepting one.*
  - *There exists a polynomial $p(n)$ such that for each $\mathcal{C} \in \mathbf{Comp}(\Pi)$ we have $|f(\mathcal{C})| \leq p(|\mathcal{C}|)$.*

Next, for every recognizer tissue P system with symport/antiport rules we design a basic recognizer P systems efficiently simulating it, according to Definition 17.

Definition 18. *Let $\Pi = (\Gamma, \Sigma, \Omega, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}, i_{in})$ be a recognizer tissue P system of degree $q \geq 1$ with communication rules and without cell division. Let us consider the basic recognizer P system $S(\Pi) = (\Gamma', \Sigma', \mu, \mathcal{M}'_1, \mathcal{R}', i'_{in})$ defined as follows:*

- *$\Gamma' = \{(a,i) : a \in \Gamma \wedge i \in \{1, \dots, q\}\} \cup \{(a,0) : a \in \Gamma \setminus \Omega\} \cup \{\mathtt{yes}, \mathtt{no}\}$.*
  The objects of $S(\Pi)$ are ordered pairs encoding objects of Π and cells where the objects are placed. From the environment, we only consider objects with finite multiplicity, that is, belonging to $\Gamma \setminus \Omega$.
- *$\Sigma' = \{(a, i_{in}) : a \in \Sigma\}$.*
- *$\mu = [\;]_1$.*
- *$\mathcal{M}'_1 = \displaystyle\sum_{i=1}^{q} \sum_{a \in \Gamma \setminus \Sigma} (a,i)^{\mathcal{M}_i(a)}$.*
  For each cell i of Π and for each object $a \in \Gamma \setminus \Sigma$ belonging to that cell, we consider in the membrane of $S(\Pi)$ the pair (a,i) with the same multiplicity.
- *In the set $\mathcal{R}'$ the following rules associated with $S(\Pi)$ are included:*
  - *For each rule $r_{\Pi} \equiv (i, a_1 \dots a_m \,/\, b_1 \dots b_n, j) \in \mathcal{R}$ with $i, j \neq 0$, associated with Π, we consider the following rule (denoted by $r_{S(\Pi)}$)*
    $(a_1,i) \dots (a_m,i)(b_1,j) \dots (b_n,j) \rightarrow (b_1,i) \dots (b_n,i)(a_1,j) \dots (a_m,j)$
  - *For each rule $r_{\Pi} \equiv (i, a_1 \dots a_m \,/\, b_1 \dots b_n, 0) \in \mathcal{R}$ with $i \neq 0$, associated with Π, we consider the following rule (denoted by $r_{S(\Pi)}$)*
    $(a_1,i) \dots (a_m,i)(b_1,0) \dots (b_s,0) \rightarrow (b_1,i) \dots (b_n,i)(a_1,0) \dots (a_r,0)$
    where $a_1, \dots, a_r, b_1, \dots, b_s \notin \Omega$ and $a_{r+1}, \dots, a_m, b_{s+1}, \dots, b_n \in \Omega$.
  - *For each rule $r_{\Pi} \equiv (0, a_1 \dots a_m \,/\, b_1 \dots b_n, i) \in \mathcal{R}$ with $i \neq 0$, associated with Π, we consider the following rule (denoted by $r_{S(\Pi)}$)*
    $(a_1,0) \dots (a_r,0)(b_1,i) \dots (b_n,i) \rightarrow (b_1,0) \dots (b_s,0)(a_1,i) \dots (a_m,i)$
    where $a_1, \dots, a_r, b_1, \dots, b_s \notin \Omega$ and $a_{r+1}, \dots, a_m, b_{s+1}, \dots, b_n \in \Omega$.

  - $(\texttt{yes}, 0) \rightarrow (\texttt{yes}, out)$; $(\texttt{no}, 0) \rightarrow (\texttt{no}, out)$.
    These rules translate the answer provided by the system Π to an answer for the system $S(\Pi)$.
- $i'_{in} = 1$, *that is, the membrane of the system is the input membrane.*

Proposition 20. *Let Π be a recognizer tissue P system with communication rules and without cell division. The system $S(\Pi)$ is a basic recognizer P system that efficiently simulates Π.*

This result provides us a limitation concerning the efficiency of tissue P systems with communication rules and without cell division. Within this framework, it is only possible to efficiently solve tractable problems, that is, problems belonging to the complexity class **P** [6].

Theorem 7. $\mathbf{P} = \mathbf{PMC}_{\mathcal{TC}}$.

7 Conclusions

In this paper, we have described the basic concepts and the main results that pertain to pioneering computational complexity in the membrane computing field.

We conclude by presenting new research directions within membrane computing complexity theory by listing some of the current open questions.

(A) Are there significant differences between uniform and semi–uniform solutions? Namely, is there some class $\mathcal{R}$ of recognizer P systems such that the inclusion $\mathbf{PMC}_{\mathcal{R}} \subseteq \mathbf{PMC}^*_{\mathcal{R}}$ is strict?

(B) Efficient uniform solutions to **NP**–complete problems have been given by models of $\mathcal{AM}(-n)$. Is it possible to efficiently solve **PSPACE**–complete problems by using families of P systems from $\mathcal{AM}(-n)$?

(C) What is the efficiency of P systems with active membranes and electrical charges where evolution and communication rules are forbidden? Are there any relations with the results obtained for polarizationless P systems?

(D) Dissolution rules provide a borderline between tractability and intractability in the framework of polarizationless P systems with active membranes making use of division rules for elementary and non–elementary membranes. What happens if division for only elementary membranes is allowed? Is $\mathbf{P} = \mathbf{PMC}^{[*]}_{\mathcal{AM}^0(+d,-n,+e,+c)}$ true?

(E) It is well known that $\mathbf{PSPACE} \subseteq \mathbf{PMC}^*_{\mathcal{AM}^0(+d,+nsr,-e,-c)}$. Determine an upper bound for that membrane computing complexity class.

(F) It is known that $\mathbf{P} = \mathbf{PMC}_{TDC(1)}$ and $\mathbf{NP} \cup \textbf{co-NP} \subseteq \mathbf{PMC}_{TDC(3)}$. What is the complexity class $\mathbf{PMC}_{TDC(2)}$? In the solution provided in [7], antiport rules of length at most 3 were used. Would it be possible to provide another efficient solution in which all rules of length 3 were symport?

Acknowledgements

The authors acknowledge the support of the project TIN2006–13425 of the Ministerio de Educación y Ciencia of Spain, cofinanced by FEDER funds, and the support of the Project of Excellence with *Investigador de Reconocida Valía* of the Junta de Andalucía, grant P08-TIC-04200.

References

1. A. Alhazov, M.J. Pérez–Jiménez: Uniform solution of QSAT using polarizationless active membranes. *Lecture Notes in Computer Science*, **4664** (2007), 122-133.
2. A. Alhazov, R. Freund, Gh. Păun: P systems with active membranes and two polarizations. *Second Brainstorming Week on Membrane Computing* (Gh. Păun et al. eds.), Report RGNC 01/04, 2004, pp. 20–35.
3. A. Alhazov, L. Pan, Gh. Păun: Trading polarizations for labels in P systems with active membranes. *Acta Informaticae*, **41**, 2-3 (2004), 111-144.
4. A. Alhazov, C. Martín–Vide, L. Pan: Solving graph problems by P systems with restricted elementary active membranes. *Lecture Notes in Computer Science*, **2950** (2004), 1–22.
5. A. Alhazov, C. Martín–Vide, L. Pan: Solving a PSPACE–complete problem by recognizing P systems with restricted active membranes. *Fundamenta Informaticae*, **58** (2003), 67–77.
6. D. Díaz–Pernil, M.A. Gutiérrez–Naranjo, M.J. Pérez-Jiménez, A. Romero–Jiménez: Efficient Simulation of Tissue–like P Systems by Transition Cell–like P systems. *Natural Computing*, online version http://dx.doi.org/10.1007/s11047-008-9102-z.
7. D. Díaz–Pernil, M.J. Pérez–Jiménez, A. Riscos–Núñez and A. Romero–Jiménez. Computational Efficiency of Cellular Division in Tissue-like Membrane Systems. *Romanian Journal of Information Science and Technology*, **11**, 3 (2008), 229–241.
8. M.R. Garey, D.S. Johnson: *Computers and Intractability. A guide to the theory of NP-completeness.* W.H. Freeman and Company, New York, 1979.
9. R. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, M. Rius–Font. Characterizing tractability by tissue–like P systems. In R. Gutiérrez–Escudero, M.A. Gutiérrez–Naranjo, Gh. Păun, I. Pérez–Hurtado and A. Riscos Núñez (eds.) *Proceedings of the Seventh Brainstorming Week on Membrane Computing*, Fénix Editora, Seville, 2009, pp. 169–180.
10. M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Núñez: A fast P system for finding a balanced 2-partition. *Soft Computing*, **9**, 9 (2005), 673–678.
11. 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. *Lecture Notes in Computer Science*, **3850** (2006), 224–240.
12. M.A. Gutiérrez–Naranjo, M.J. Pérez–Jiménez, A. Riscos–Núñez, F.J. Romero–Campero, A. Romero–Jiménez: Characterizing tractability by cell–like membrane systems. In K.G. Subramanian, K. Rangarajan, M. Mukund (eds.) *Formal models, languages and applications*, World Scientific, Singapore, 2006, pp. 137–154.
13. S.N. Krishna, R. Rama: A variant of P systems with active membranes: Solving NP–complete problems. *Romanian Journal of Information Science and Technology*, **2**, 4 (1999), 357–367.

14. A. Leporati, C. Ferretti, G. Mauri, M.J. Pérez–Jiménez, C. Zandron: Complexity aspects of polarizationless membrane systems. Submitted 2008.
15. N. Murphy, D. Woods: Active membrane systems without charges and using only symetric elementary division characterise **P**. *Lecture Notes in Computer Science*, **4860** (2007), 367–384.
16. A. Obtulowicz: Deterministic P systems for solving SAT problem. *Romanian Journal of Information Science and Technology*, **4**, 1–2 (2001), 551–558.
17. L. Pan, A. Alhazov, T.-O. Ishdorj: Further remarks on P systems with active membranes, separation, merging, and release rules. *Second Brainstorming Week on Membrane Computing* (Gh. Păun et al. eds.), Report RGNC 01/04, 2004, pp. 316–324.
18. L. Pan, T.-O. Ishdorj: P systems with active membranes and separation rules. *Journal of Universal Computer Science*, **10**, 5 (2004), 630–649.
19. A. Păun: On P systems with membrane division. In I. Antoniou, C.S. Calude, M.J. Dinneen (eds.) *Unconventional Models of Computation*, Springer, London, 2000, pp. 187–201.
20. Gh. Păun: Computing with membranes. *Journal of Computer and System Sciences*, **61**, 1 (2000), 108–143, and *Turku Center for CS-TUCS Report* No. 208, 1998
21. Gh. Păun: *Membrane Computing. An introduction.* Springer-Verlag, Berlin, 2002.
22. Gh. Păun: P systems with active membranes: Attacking **NP**–complete problems. *Journal of Automata, Languages and Combinatorics*, **6**, 1 (2001), 75–90.
23. Gh. Păun: Computing with membranes: Attacking **NP**–complete problems. In I. Antoniou, C.S. Calude, M.J. Dinneen (eds.) *Unconventional Models of Computation*, Springer, London, 2000, pp. 94–115.
24. Gh. Păun: Further twenty six open problems in membrane computing. *Third Brainstorming Week on Membrane Computing* (M.A. Gutiérrez et al. eds.), Fénix Editora, Sevilla, 2005, pp. 249–262.
25. M.J. Pérez–Jiménez: An approach to computational complexity in Membrane Computing. *Lecture Notes in Computer Science*, **3365** (2005), 85–109.
26. M.J. Pérez-Jiménez, A. Riscos-Núñez: Solving the Subset-Sum problem by active membranes. *New Generation Computing*, **23**, 4 (2005), 367–384.
27. M.J. Pérez-Jiménez, A. Riscos-Núñez: A linear–time solution to the Knapsack problem using P systems with active membranes. *Lecture Notes in Computer Science*, **2933** (2004), 250–268.
28. M.J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini: A polynomial complexity class in P systems using membrane division. *Journal of Automata, Languages and Combinatorics*, **11**, 4 (2006), 423-434.
29. M.J. Pérez-Jiménez, F.J. Romero–Campero: Attacking the Common Algorithmic Problem by recognizer P systems. *Lecture Notes in Computer Science*, **3354** (2005), 304–315.
30. M.J. Pérez–Jiménez, F.J. Romero–Campero: An efficient family of P systems for packing items into bins. *Journal of Universal Computer Science*, **10**, 5 (2004), 650–670.
31. M.J. Pérez–Jiménez, F.J. Romero-Campero: Trading polarizations for bi-stable catalysts in P systems with active membranes. *Lecture Notes in Computer Science*, **3365** (2005), 373–388.
32. M.J. Pérez–Jiménez, A. Romero–Jiménez, F. Sancho–Caparrini: Complexity classes in cellular computing with membranes. *Natural Computing*, **2**, 3 (2003), 265–285.
33. A. E. Porreca: *Computational Complexity Classes for Membrane Systems*, Master Degree Thesis, Universita' di Milano-Bicocca, Italy, 2008.

34. A.E. Porreca, G. Mauri, C. Zandron: Complexity classes for membrane systems. *Informatique théorique et applications*, **40**, 2 (2006), 141–162.
35. A. Riscos–Núñez: *Cellular Programming: efficient resolution of NP–complete numerical problems.* PhD. Thesis, University of Sevilla, Spain, 2004.
36. P. Sosík, A. Rodríguez–Patón: Membrane computing and complexity theory: A characterization of PSPACE. *Journal of Computer and System Sciences*, **73** (2007), 137–152.
37. P. Sosík: The computational power of cell division. *Natural Computing*, **2**, 3 (2003), 287–298.
38. D. Woods, N. Murphy, M.J. Pérez-Jiménez, A. Riscos-Núñez: Membrane disoolution and division in **P**. *Lecture Notes in Computer Science*, **5715** (2009), 263-277.
39. C. Zandron, C. Ferretti, G. Mauri: Solving NP–complete problems using P systems with active membranes. In I. Antoniou, C.S. Calude, M.J. Dinneen (eds.) *Unconventional Models of Computation*, Springer, Berlin, 2000, pp. 289–301.
40. C. Zandron, A. Leporati, C. Ferretti, G. Mauri, M.J. Pérez–Jiménez: On the computational efficiency of polarizationless recognizer P systems with strong division and dissolution. *Fundamenta Informaticae*, **87**, 1 (2008), 79-91.