Dependency Graph Technique Revisited

Luis Valencia-Cabrera, David Orellana-Martín,
Ignacio Pérez-Hurtado, 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
{lvalencia,dorellana,perezh,marper}@us.es

Summary. The dependency graph technique was initially thought as a method to find short paths in the computation tree of a membrane system using *weak metrics*. It could be used to obtain reasonably fast SAT-solvers, capable of competing with the ones available in the literature. Later on, they were used as a method to demonstrate the non-efficiency of some membrane systems, capturing the dynamics of the systems by a static directed graph structure. Recently, the dependency graphs have also been used to establish negative results in Membrane Computing. Specifically, in this work, demonstrating the inability of a kind of membrane system to solve some decision problems efficiently by means of a single system.

1 Introduction

The *computational efficiency* of a model in a computing paradigm refers to its ability to provide polynomial time solutions for computationally hard problems, generally achieved by making use of an exponential workspace constructed in a natural way. Aspects related to the computational efficiency within membrane computing were first analyzed in 1999, with the introduction of a new computing model called *P system with active membranes* [5]. These systems are non-cooperative (the left hand side of any rule consists of only one object) and their membranes play a relevant role in computations to the extent that new membranes can be created by division rules. The membranes of these systems are supposed to have one of three possible electrical polarizations: positive, negative or neutral. In this context, it was given an *ad-hoc* solution to the boolean satisfiability problem (SAT) by means of such kind of P systems. More specifically, a P system with active membranes which makes use of *simple* object evolution rules (only one object is produced for this kind of rules), dissolution rules and division rules for elementary and non-elementary membranes, is associated with every instance φ of SAT. Thus, the syntactic structure of the formula is "captured" by the description of the system and, furthermore, in this context a P system can only process one instance of

the problem. The solution provided runs in linear time with respect to the *size* of the input formula φ, that is, the maximum between the number of variables and the number of clauses in φ.

Usually, computational complexity theory deals with decision problems, that is, problems requiring a `yes/no` answer. Each decision problem has a language associated with it, in a natural way, so that solving such problems is defined through the recognition of the corresponding language. Thus, in order to describe in a formal way what solving a decision problem means, basic *recognizer transition P systems* (initially called *decision P systems*) were defined [7].

Let us recall that an abstract problem can be solved by using a single Turing machine, that is, for every instance of the problem, the Turing machine receiving the input corresponding to that instance returns the correct answer. This is due to the fact that these machines have an unlimited and unrestricted memory, given the infinite tape it includes (consisting of an infinite number of cells). Bearing in mind that the ingredients necessary to define a membrane system are finite, an abstract problem should be solved, in general, by an infinite numerable family of membrane systems, in such a way that each system in the family is in charge of processing all the instances having the same size.

The concept of solvability of decision problems in polynomial time in a uniform way by means of a family of recognizer membrane systems was introduced in [7].

Definition 1. *Let $X = (I_X, \theta_X)$ be a decision problem and let $\mathcal{R}$ be a class of recognizer membrane systems with input membrane. Let $\Pi + u$ be the P system Π with the multiset u in the input membrane in the initial configuration. We say that X is* solvable in polynomial time *and* uniform *way by a family $\{\Pi(u) \mid u \in I_X\}$ of systems from $\mathcal{R}$, denoted by $X \in \mathbf{PMC}_{\mathcal{R}}$, if the following holds:*

- *The family is* polynomially uniform by Turing machines*, that is, there exists a deterministic Turing machine working in polynomial time which constructs the system $\Pi(n)$ from the number $n \in \mathbb{N}$, expressed in unary.*
- *There exists a pair (cod, s) of polynomial–time computable functions over I_X such that for each $n \in \mathbb{N}$, the set $s^{-1}(n)$ is finite, and for each $u \in I_X$, $s(u) \in \mathbb{N}$ and $cod(u)$ is an input multiset of the system $\Pi(s(u))$.*
- *The family is* polynomially bounded with respect to (X, cod, s)*; that is, there exists $k \in \mathbb{N}$ such that for each $u \in I_X$, every computation of the system $\Pi(s(u)) + cod(u)$ performs at most $|u|^k$ steps.*
- *The family is* sound with respect to (X, cod, s)*, that is, for each $u \in I_X$, if* there exists *an accepting computation of $\Pi(s(u)) + cod(u)$ then $\theta_X(u) = 1$.*
- *The family is* complete with respect to (X, cod, s)*, that is, for each $u \in I_X$, if $\theta_X(u) = 1$ then* every *computation of $\Pi(s(u)) + cod(u)$ is an accepting computation.*

According with the previous definition:

– We say that the family $\{\Pi(n) \mid n \in \mathbb{N}\}$ provides a *uniform solution* to the problem X and the ordered pair (cod, s) is a *polynomial encoding* from the problem X to the family $\{\Pi(n) : n \in \mathbb{N}\}$.

– For each instance $u \in I_X$, the system $\Pi(s(u))$ processes u when the input multiset $cod(u)$ is supplied to the corresponding input membrane. Besides, the system $\Pi(s(u)) + cod(u)$ is *confluent*, in the sense that all computations must give the same answer (either all computations are accepting computations or all computations are rejecting computations).

As a direct consequence of working with recognizer membrane systems, these complexity classes are closed under complement. Moreover, it is easy to prove that they are closed under polynomial–time reductions [8].

In the previous definition, uniform solutions for decision problems by means of families of membrane systems have been introduced. It seems interesting to analyze what kind of membrane systems are capable of solving decision problems through only one unique system. In this context, it is essential to clarify how the instances of the problem are introduced into the system. Next, we consider the case in which the instances are directly introduced inside the system (*free* of resources) by means of a representation of the problem to be solved. It is important to remark that this means that the input alphabet of the P system is the same one that the alphabet of the problem, so there is no possibility of encoding, for instance, an instance of a problem from **P** to an object `yes` or an object `no`.

Definition 2. *Let $X = (I_X, \theta_X)$ be a decision problem where I_X is a language over a finite alphabet Σ_X. Let $\mathcal{R}$ be a class of recognizer membrane systems with input membrane. We say that problem X is solvable in polynomial time by a single membrane system Π from $\mathcal{R}$, free of resources, denoted by $X \in \mathbf{PMC}_{\mathcal{R}}^{1f}$, if the following holds:*

- *The input alphabet of Π is Σ_X.*
- *The system Π is* polynomially bounded *with regard to X; that is, there exists a polynomial $p(r)$ such that for each instance $u \in I_X$, every computation of the system Π with input multiset u performs at most $p(|u|)$ steps.*
- *The system Π is* sound *with regard to X; that is, for each instance $u \in I_X$, if there exists an accepting computation of the system Π with input multiset u then $\theta_X(u) = 1$.*
- *The system Π is* complete *with regard to X; that is, for each instance $u \in I_X$ such that $\theta_X(u) = 1$, every computation of the system Π with input multiset u is an accepting computation.*

From the previous definition it is easy to prove that $\mathbf{PMC}_{\mathcal{R}}^{1f} \subseteq \mathbf{PMC}_{\mathcal{R}}$, for every class $\mathcal{R}$ of recognizer membrane systems with input membrane.

2 Dependency Graph Associated with P Systems. The Origin

The dynamics of a membrane system provides, in a natural way, a tree of computation. More precisely, the *computation tree* of a membrane system Π, denoted $Comp(\Pi)$, is a rooted labelled maximal tree defined as follows:

- Its nodes are labelled by the configurations of Π.
- Its edges are labelled by applicability matrices (maximal multiset of rules applicable to a configuration).
- The root of the tree is the initial configuration of Π.
- The children of a node labelled by $\mathcal{C}$ are the configurations $\mathcal{C}'$ which can be obtained from $\mathcal{C}$ in one transition step.

The maximal branches of $Comp(\Pi)$ will be called *computations* of Π. A computation of Π is a halting computation if and only if it is a finite branch. The labels of the leaves of $Comp(\Pi)$ are called *halting configurations*.

Given a semi-uniform or uniform solution (in polynomial time) for a decision problem by means of a family of recognizer membrane systems, every instance of the problem is processed by a system of the family. This system must be confluent, so in order to know its answer for any instance it is enough to consider only one computation of such system. In this context, an exciting challenge would be looking for a computation with minimum length. For that, it would be interesting to analyze the degree of closeness between two configurations. The problem is especially hard if we want to quantify that proximity in order to make useful comparisons. Some *weak metrics* on configurations of a membrane system with a fixed structure of membranes have been studied in [2]. In this context, in order to search for the shortest paths in a graph providing a *sound* computation of the system, the *dependency graph* associated with the set of rules of a recognizer membrane system was introduced. This concept is based on the dependence among elements of the alphabet with respect to the set of rules of the P system. Several weak metrics over the set of configurations of the system based on the concept of dependency graph were considered, starting from the notion of *distance* between two nodes of the graph (the length of the shortest path connecting v_1 and v_2, or *infinite* if there is no path from v_1 to v_2).

3 Dependency Graph as a Technique to Prove the Non-Efficiency of Membrane Systems

In some kind of recognizer membrane systems, it is possible to consider a directed graph (also called *dependency graph*) verifying the following properties: (a) it can be constructed from the set or rules of the system in polynomial time, that is, in a time bounded by a polynomial function depending on the total number of rules and the maximum length of them; and (b) the accepting computations of such systems can be characterized by means of a "reachability" property in the dependency graph associated with it (the existence of a path in the graph between two specific nodes). Therefore, dependency graphs provide a technique to tackle the limits on efficient computations in membrane systems; that is, the non-efficiency of such systems.

As an illustration example, recognizer polarizationless P systems with active membranes not using dissolution rules are considered. The rules of such systems

can be considered as a *dependency* relation between the object triggering the rule and the objects produced by its application. For instance,

- Object evolution rules $[\, a \to u \,]_h$ can be described as follows: the pair (a, h) produces the pair (b, h), for each $b \in supp(u)$, being $supp(u) = \{x \in \Gamma \mid f(x) > 0\}$.
- Send-in communication rules $a[\ \ ]_h \to [\, b \,]_h$ can be described as follows: the pair $(a, p(h))$ produces the pair (b, h).
- Send-out communication rules $[\, a \,]_h \to [\ \ ]_h\ b$ can be described as follows: the pair (a, h) produces the pair $(b, p(h))$.
- Division rules for elementary membranes $[\, a \,]_h \to [\, b \,]_h\ [\, c \,]_h$ can be described as follows: the pair (a, h) produces the pairs (b, h) and (c, h).

Let us recall that if h is the label of a membrane, then $p(h)$ denotes the label of the parent of such membrane labelled with h. We adopt the convention that the father of the skin membrane (the root of the tree) is the environment and its label is denoted by env.

In this context, division rules for non-elementary membranes do not provide any information. These ideas can be formalized as follows:

Definition 3. *Let $\Pi = (\Gamma, H, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R})$ be a recognizer polarizationless P system with active membranes of degree $q \geq 1$ not using dissolution rules. Let us assume that the label "env" of the environment of Π is in H. Let $p(h)$ be the label of the parent membrane of the membrane labelled by h. Let $ch(h)$ be the set of labels of the child membranes of the membrane labelled by h. The dependency graph associated with Π is the directed graph $G_\Pi = (V_\Pi, E_\Pi)$ defined as follows:*

- *The set of vertices is $V_\Pi = \{s_\Pi\} \cup VL_\Pi \cup VR_\Pi$, where $s_\Pi \notin \Gamma \times H$ and:*
  $VL_\Pi = \{(a,h) \in \Gamma \times H \mid \exists u \in M(\Gamma)\ ([a \to u]_h \in \mathcal{R}) \vee$
  $\exists b \in \Gamma\ ([a]_h \to [\ ]_h b \in \mathcal{R}) \vee$
  $\exists b \in \Gamma\ \exists h' \in ch(h)\ (a[\ ]_{h'} \to [b]_{h'} \in \mathcal{R}) \vee$
  $\exists b, c \in \Gamma\ ([a]_h \to [b]_h[c]_h \in \mathcal{R}))\}$.
  $VR_\Pi = \{(b,h) \in \Gamma \times H \mid \exists a \in \Gamma\ \exists u \in M(\Gamma)\ ([a \to u]_h \in \mathcal{R} \wedge b \in supp(u)) \vee$
  $\exists a \in \Gamma\ \exists h' \in ch(h)\ ([a]_{h'} \to [\ ]_{h'} b \in \mathcal{R}) \vee$
  $\exists a \in \Gamma\ (a[\ ]_h \to [b]_h \in \mathcal{R}) \vee$
  $\exists a, c \in \Gamma\ ([a]_h \to [b]_h[c]_h \in \mathcal{R} \vee [a]_h \to [c]_h[b]_h \in \mathcal{R})\}$.
- *The set of arcs is $E_\Pi = E^1_\Pi \cup E^2_\Pi$, where:*
  $E^1_\Pi = \{(s_\Pi, (a,h)) \mid h \in H \setminus \{env\} \wedge a \in \mathcal{M}_h\}$
  $E^2_\Pi = \{((a,h),(b,h')) \mid \exists u \in M(\Gamma)\ ([a \to u]_h \in \mathcal{R} \wedge b \in supp(u) \wedge h = h') \vee$
  $([a]_h \to [\ ]_h b \in \mathcal{R}\ \wedge\ h' = p(h)) \vee$
  $(a[\ ]_{h'} \to [b]_{h'} \in \mathcal{R}\ \wedge h\ = p(h')) \vee$
  $\exists c \in \Gamma\ ([a]_h \to [b]_h[c]_h \in \mathcal{R} \wedge h = h') \vee$
  $\exists c \in \Gamma\ ([a]_h \to [c]_h[b]_h \in \mathcal{R} \wedge h = h')\}$.

The node s_Π is called the source node of G_Π. The node $(\mathtt{yes}, env)$ is called the accepting node. The node $(\mathtt{no}, env))$ is called the rejecting node of G_Π.

Bearing in mind that all computations of a recognizer P system halt, we deduce that each path in the dependency graph associated with it must be simple, that is, all vertices in the path are different.

3.1 Non-efficiency of a kind of membrane systems

At the beginning of 2005, Gh. Păun (problem **F** from [6]) wrote: *My favorite question (related to complexity aspects in P systems with active membranes 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.*

Let us denote by $\mathcal{AM}^0(\alpha, \beta)$ the class of all polarizationless recognizer P systems with active membranes such that: (a) if $\alpha = +d$ (resp., $\alpha = -d$) then dissolution rules are permitted (resp., forbidden); and (b) if $\beta = +ne$ (resp., $\beta = -ne$) then division rules for elementary and non-elementary membranes (resp., only for elementary membranes) are permitted. Then, the so-called *Păun's conjecture* can be formally written in terms of membrane computing complexity classes as follows: $\mathbf{PMC}_{\mathcal{AM}^0(+d,-ne)} = \mathbf{P}$.

An *affirmative answer* to this conjecture would indicate that the ability to create an exponential amount of workspace (expressed in terms of the number of membranes and objects) in polynomial time, is not enough in order to solve computationally hard problems efficiently. On the other hand, assuming that $\mathbf{P} \neq \mathbf{NP}$, a *negative answer* to the conjecture would show that division rules for elementary membranes provide a borderline between the *non-efficiency* (only problems in class **P** can be solved in polynomial time in a computing model) and the *presumed efficiency* (ability of a computing model to provide polynomial time solutions for **NP**-complete problems) of polarizationless P systems with active membranes.

On the other hand, when dissolution is not allowed, the dependency graph associated with a P system from $\mathcal{AM}^0(-d, +ne)$ can be constructed by a deterministic Turing machine working in polynomial time (see [3] for details). Moreover, dependency graphs can be used to characterize the behavior of the system through the analysis of simple paths. Specifically, given a recognizer P system Π from $\mathcal{AM}^0(-d, +ne)$, there exists an accepting computation of Π if and only if there exists a simple path in the dependency graph G_Π from the source node s_Π to the accepting node $(\texttt{yes}, env)$ with length greater than or equal to 2 (see [3] for details).

Let $\{\Pi(n) \mid n \in \mathbb{N}\}$ be a family of P systems from $\mathcal{AM}^0(-d, +ne)$ solving a decision problem $X = (I_X, \theta_X)$ in polynomial time. Let (cod, s) be a polynomial encoding associated with that solution. Then, for each instance $u \in I_X$, the answer of the problem is `yes`, *i.e.* $\theta_X(u) = \texttt{yes}$, if and only if there exists a simple path in the dependency graph associated with $\Pi' = \Pi(s(u)) + cod(u)$, from the source node $s_{\Pi'}$ to the accepting node $(\texttt{yes}, env)$.

Theorem 1. $\mathbf{PMC}_{\mathcal{AM}^0\,(-d,+ne)} = \mathbf{P}$.

Proof. On the one hand, it is well known that if $\mathcal{R}$ is a class of recognizer membrane systems then $\mathbf{P} \subseteq \mathbf{PMC}_{\mathcal{R}}$. On the other hand, in order to show that $\mathbf{PMC}_{\mathcal{AM}^0(-d,+ne)} \subseteq \mathbf{P}$, let $X \in \mathbf{PMC}_{\mathcal{AM}^0(-d,+ne)}$ and let $\{\Pi(n) \mid n \in \mathbb{N}\}$ be a family of P systems from $\mathcal{AM}^0(-d,+ne)$ solving X in polynomial time in a uniform way. Let (cod, s) be a polynomial encoding associated with that solution. Let us see that there exists a polynomial time reduction from X to the `REACHABILITY` problem. The `REACHABILITY` is the following decision problem: *given a directed graph $G = (V, E)$ with two specified vertices s and t, determine if there is a path from s to t.* Some algorithms to solve this problem (*e.g.* BFS or DFS) determine if two vertices are connected in $O(\max(|V|, |E|))$ time, using $O(|V|)$ space. This space can be reduced to $O(\log^2|V|)$ by using an MFS (see [4], pp. 149-150). Thus, `REACHABILITY` $\in \mathbf{P}$. For that, let us consider the mapping F from I_X to the set of instances $I_{\texttt{REACHABILITY}}$ defined as follows: $F(u) = (G_{\Pi(s(u))+cod(u)}, s_{\Pi(s(u))+cod(u)}, (yes, env))$, for each instance u of X. Then, F is a polynomial time computable function such that

$$u \in L_X \Longleftrightarrow F(u) \in L_{\texttt{REACHABILITY}}$$

Finally, we deduce that $X \in \mathbf{P}$ because the class $\mathbf{P}$ is closed under polynomial–time reductions, $X \leq^p$ `REACHABILITY` and `REACHABILITY` $\in \mathbf{P}$.

Hence, in the framework of polarizationless P systems with active membranes not using dissolution rules we have a *partial affirmative* answer to Păun's conjecture, that is, $\mathbf{PMC}_{\mathcal{AM}^0(-d,+ne)} = \mathbf{P}$. The answer is partial because dissolution rules have been forbidden.

In [1], it was given a uniform linear time solution to the `QBF-SAT` problem, a well known **PSPACE**-complete problem [4], by means of a family of recognizer polarizationless P systems with active membranes using dissolution rules and division for elementary and non–elementary membranes. Thus, $\mathbf{PSPACE} \subseteq \mathbf{PMC}_{\mathcal{AM}^0(+d,+ne)}$. Hence, assuming that $\mathbf{P} \neq \mathbf{NP}$, we have a *partial negative* answer to Păun's conjecture: computationally hard problems can be efficiently solved avoiding polarizations. The answer is partial because division rules for non–elementary membranes have been required.

4 Dependency Graph as a Technique to Prove Negative Results in Membrane Computing

Let $\mathcal{R}$ be a class of recognizer membrane systems such that every system from $\mathcal{R}$ is associated with a dependency graph verifying the following property: a computation of a system from $\mathcal{R}$ is an accepting computation if and only if there exists a path between two distinguished nodes in the dependency graph associated with the system. In this situation, it is possible to show that some decision problem $X = (I_X, \theta_X)$ *cannot be solved* in polynomial time in a uniform way by means of a

single membrane system, free of resources, from $\mathcal{R}$. This remark is illustrated by an example.

The ONLY-ONE-OBJECT problem is the decision problem $X = (I_X, \theta_X)$ defined as follows: $I_X = \{a^n \mid n \in \mathbb{N}, n \geq 1\}$ and $\theta_X(a^n) = 1$ if and only if $n = 1$. It is easy to design a deterministic Turing machine which takes two computation steps, solving the ONLY-ONE-OBJECT problem. Let us see that ONLY-ONE-OBJECT $\notin \mathbf{PMC}^{1f}_{\mathcal{AM}^0(-d,+ne)}$.

Theorem 2. *There is no recognizer membrane system from the class* $\mathcal{AM}^0(-d, +ne)$ *solving the* ONLY-ONE-OBJECT *problem in polynomial time by a single membrane system and free of resources.*

Proof. Let us assume that there exists a recognizer membrane system Π from $\mathcal{AM}^0(-d, +ne)$ verifying the following: (a) the input alphabet of Π is the singleton $\{a\}$; (b) every computation of Π with input multiset $\{a\}$ is an accepting computation; and (c) every computation of Π with input multiset $\{a^n\}$, for each $n > 1$, is a rejecting computation.

Let us denote by $G_{\Pi+\{a\}}$ (respectively, $G_{\Pi+\{a^n\}}$, for each $n > 1$) the dependency graph associated with the system $\Pi + \{a\}$ (resp. $\Pi + \{a^n\}$). Then, for each $n > 1$, we have $G_{\Pi+\{a\}} = G_{\Pi+\{a^n\}}$, since there would always be an edge $(s_\Pi, (a, i_{in}))$ in the dependency graph, and the rest of the graph would remain the same. Besides, every computation of $\Pi + \{a\}$ is an accepting computation if and only if every computation of $\Pi + \{a^n\}$, for each $n > 1$, is an accepting computation, which is a contradiction of the initial hypothesis, thus there cannot exist such membrane system.

5 Conclusions

Along this work, some of the main results concerning the use of dependency graphs within membrane computing to analyze the computational efficiency of computing models have been reviewed. It is worth pointing out that, albeit the **P** *versus* **NP** problem is the most important one in Computer Science, there are other interesting problems in the field of Computational Complexity Theory, also below **P**. When using polynomial precomputed resources, problems from **P** can be easily solved. But considering membrane systems *free of precomputed resources*, things change in such a way that there cannot be trivial solutions that could be obtained at first. Then, it would be useful to study these kinds of systems to solve problems below this complexity class. In this case, with this technique it has been demonstrated that there is no solution to the ONLY-ONE-OBJECT problem by means of a single membrane system from $\mathcal{AM}^0(-d, +ne)$.

Adapting currently used methodologies to new applications is an interesting future research line to improve existing results and obtain new ones. Besides, the search for new techniques to demonstrate the non-efficiency or the inability for certain membrane systems to solve some decision problems is critical when addressing

the **P** *versus* **NP** and other interesting problems in the field of Computational Complexity Theory.

Acknowledgements

This work was supported in part by the research 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. A. Alhazov, M.J. Pérez-Jiménez. Uniform solution to QSAT using polarizationless active membranes. In J. Durand-Lose, M. Margenstern (eds.) *Machines, Computations, and Universality 5th International Conference, MCU 2007, Orléans, France, September 10-13, 2007. Proceedings. Lecture Notes in Computer Science*, **4664** (2007), 122-133.
2. A. Cordón-Franco, M. A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Núñez. Weak metrics on configurations of a P system. In Gh. Paun, A. Riscos, Á. Romero, F. Sancho (eds.) *Proceedings of the Second Brainstorming Week on Membrane Computing*, Report RGNC 01/2004, 2004, pp. 139-151.
3. 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, Gh. Păun, G. Rozenberg, A. Salomaa (eds.) *Membrane Computing, 6th International Workshop, WMC 2005, Vienna, Austria, July 18-21, 2005, Revised Selected and Invited Papers. Lecture Notes in Computer Science*, **3850** (2006), 224-240.
4. C.H. Papadimitriou. *Computational Complexity*, Addison–Wesley, Massachusetts, 1995.
5. Gh. Păun. P systems with active membranes: attacking **NP**–complete problems, *Journal of Automata, Languages and Combinatorics*, **6**, 1 (2001), 75-90. A preliminary version in *Centre for Discrete Mathematics and Theoretical Computer Science, CDMTCS Research Report Series*-102, May 1999, 16 pages.
6. Gh. Păun. Further twenty six open problems in membrane computing. In M. A. Gutiérrez-Naranjo, A. Riscos-Núñez, F.J. Romero-Campero, D. Sburlan (eds.) *Proceedings of the Third Brainstorming Week on Membrane Computing*, Fénix Editora, Sevilla, 2005, 249-262.
7. M.J. Pérez-Jiménez, Á. Romero-Jiménez, F. Sancho-Caparrini. Decision P systems and the $\mathbf{P} \neq \mathbf{NP}$ conjecture. In Gh. Păun, Gr. Rozenberg, A. Salomaa, C. Zandron (eds.) Membrane Computing 2002. *Lecture Notes in Computer Science*, **2597** (2003), 388-399. A preliminary version in Gh. Păun, C. Zandron (eds.) *Pre-proceedings of Workshop on Membrane Computing 2002*, MolCoNet project-IST-2001-32008, Publication No. 1, Curtea de Arges, Romanian, August 19-23, 2002, pp. 345-354.
8. M.J. Pérez-Jiménez, Á. 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. A preliminary version in E. Csuhaj-Varjú,

C. Kintala, D. Wotschke, Gy. Vaszil (eds.) *Proceedings of the Fifth International Workshop on Descriptional Complexity of Formal Systems*, DCFS 2003, Budapest, Hungary, July 12-14, 2003, pp. 284-294.