Computational Complexity Theory in Membrane Computing: Seventeen Years After

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
marper@us.es

Summary. In this work we revisit the basic concepts, definitions of computational complexity theory in membrane computing. The paper also discusses a novel methodology to tackle the **P** versus **NP** problem in the context of the aforementioned theory. The methodology is illustrated with a collection of frontiers of tractability for several classes of P systems.

1 Introduction

At the end of 1998, the first foundations of a new computational paradigm, called *Membrane Computing*, inspired by some basic biological features of living cells, as well as in the cooperation of cells in tissues, organs and organisms, were introduced by Gh. Păun [20]. The seminal paper was focused on the study of the *computational completeness* (i.e. to be equivalent in power to Turing machines) of the models, generically called *membrane systems*, considered: *transition P systems*, *P systems based on rewriting* and *splicing P systems*. In particular, a transition P system consists of a collection of unit processors, called *membranes*, hierarchically structured by means of a rooted tree. These membranes delimit regions containing multisets of objects which can evolve according with some prefixed rewriting rules, being applied in a non-deterministic an maximally parallel way.

Aspects related to the *computational efficiency* (i.e. the ability to provide polynomial-time solutions for computationally hard problems by making use of an exponential workspace constructed in a natural way) were first analyzed in 1999 with the introduction of a new computing model, called *P system with active membranes* [21]. 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 the three possible electrical polarizations: positive, negative or neutral. In this context, an *ad-hoc* solution

to the satisfiability problem (SAT) by means of such kind of P systems, was given. 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 provided solution runs in linear time with respect to the input length of φ, that is, the maximum of number of variables and number of clauses of the formula φ. In [33], a similar ad-hoc solution to the SAT problem by means of P system with active membranes but without dissolution rules which makes use of division rules only for elementary membranes, was provided. In this situation, different instances of the SAT problem having the same number of variables and the same and number of clauses, will be processed by different membrane systems.

In order to define in a formal way what solving a decision problem means, basic *recognizer transition P systems* (initially called *decision P systems*) were defined [26]. In this context, the *computational efficiency* of this kind of membrane systems was studied. With this respect, two interesting results were established [5]:

- Every deterministic Turing machine working in polynomial time can be simulated in polynomial time by a family of basic recognizer transition P systems.
- If a decision problem is solvable in polynomial time by a family of basic recognizer transition P systems, then there exists a deterministic Turing machine solving it in polynomial time.

Consequently, only problems in class **P** can be efficiently solved by basic recognizer transition P systems. Then, in order to provide polynomial-time solutions to computationally hard problems, it was necessary to consider new kinds of rules able to increase the number of membranes during a computation. Specifically, the membrane systems should have the ability of trading space for time by providing an exponential workspace (expressed in terms of the number of objects and the number of membranes) in polynomial (often, linear) time. In order to implement this capability in models of Membrane Computing, different mechanisms have been considered inspired by the cellular mitosis (division rules), autopoiesis (creation rules) or membrane fission (separation rules), among others.

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 with input that instance returns the correct answer. This is due to the fact that these machines have an unlimited and unrestricted memory since its tape is infinite (consists on infinite cells). Bearing in mind that the ingredients necessary to define a membrane system are finite, an abstract problem should be solved, in general, by a numerable family of membrane systems in such manner that each one of them is in charge of processing all the instances with the same size. However, some decision problems can be solved by means of a single membrane system.

2 Preliminaries

The present work is developed within the framework of P systems with active membranes. In order to make this paper self-contained, let us introduce next some preliminary concepts which will be used in the sequel.

2.1 Languages and Multisets

An *alphabet* Γ is a non-empty set whose elements are called *symbols*. A *string* u over Γ is a mapping from a natural number $n \in \mathbb{N}$ onto Γ, that is, a finite sequence over Γ. A *language* over Γ is a set of strings over Γ.

A *multiset* over an alphabet Γ is an ordered pair (Γ, f), where f is a mapping from Γ onto the set of natural numbers $\mathbb{N}$. The *support* of a multiset $m = (\Gamma, f)$ is defined as $supp(m) = \{x \in \Gamma \mid f(x) > 0\}$. A multiset is finite if its support is a finite set. We denote by $M(\Gamma)$ the set of all multisets over Γ.

Let $m_1 = (\Gamma, f_1)$, $m_2 = (\Gamma, f_2)$ be multisets over Γ, then the union of m_1 and m_2, denoted by $m_1 + m_2$, is the multiset (Γ, g), where $g(x) = f_1(x) + f_2(x)$ for each $x \in \Gamma$. The relative complement of m_2 in m_1, denoted by $m_1 \setminus m_2$, is the multiset (Γ, g), where $g(x) = f_1(x) - f_2(x)$ if $f_1(x) \geq f_2(x)$, and $g(x) = 0$ otherwise.

2.2 Decision problems

Usually, complexity theory deals with *decision problems*. A decision problem, X, is an ordered 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 over I_X. 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 $L_X = \{w \in I_X \mid \theta_X(w) = 1\}$. 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 $\theta_{X_L} = \{(x, 1) \mid x \in L\} \cup \{(x, 0) \mid x \notin L\}$. In this context, the solvability of decision problems is defined through the recognition of the languages associated with them. Let M be a (*decision*) *Turing machine* (the result of any halting computation of M is `yes` or `no`) with a working alphabet Γ and L is a language over Γ. 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 the following holds: $u \in L$ if and only if there exists a computation of M with input u such that the answer is *yes*, for any string u over Γ (we say that M *accepts* u).

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. Following [3], 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). The *size* $|u|$ of an instance u can be defined as the length of the string associated with it, in some reasonable encoding scheme.

2.3 The Graph Reachability Problem and the Circuit Value Problem

A *rooted tree* is a connected, acyclic, undirected graph in which one of the vertices (called *the root of the tree*) is distinguished from the others. Given a node x (different from the root) in a rooted tree, if the last edge on the (unique) path from the root to node x is $\{x, y\}$ (so $x \neq y$), then y is **the** *parent* of node x and x is **a** *child* of node y. We denote it by $y = p(x)$ and $x \in ch(y)$. The root is the only node in the tree with no parent. A node with no children is called a *leaf* of the rooted tree (see [1] for details).

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. We denote by L_X the set $L_X = \{w \in I_X \mid \theta_X(w) = 1\}$.

The `REACHABILITY` (or accessibility) problem is the following decision problem: *given a directed graph $G = (V, E)$ with two specified vertices s and t, determine whether or not there is a path from s to t.*

There are algorithms solving this problem, for instance, search algorithms based on breadth-first search or depth-first search. These algorithms determine whether two vertices are connected in $O(\max(|V|, |E|))$ time. Moreover, they basically need to store at most $|V|$ items, so these algorithms use $O(|V|)$ space. But this quantity of space can be reduced to $O(\log^2|V|)$ by using an algorithm that could be called middle-first search (see [19] for details, pp. 149-150). In particular, `REACHABILITY` $\in \mathbf{P}$.

The `CIRCUIT VALUE` problem is the following decision problem: *given a circuit with no variable gates and where each input gate has an associated Boolean value, determine whether or not the circuit evaluates to* `true`.

It is well known that the circuit value problem is a **P**–complete problem (see theorem 8.1 in [19] for details).

3 Recognizer Membrane Systems

The first foundations of a computational complexity theory in Membrane Computing were given in [26, 27]. In the seminal paper, the models defined are (cell-like) P systems with output membrane but without input membrane. Any such P system has only one initial configuration characterized by the initial multisets over the working alphabet. In [26], *P systems with input membrane* wee introduced. These systems have a distinguished membrane (the input membrane) and an input alphabet Σ (strictly contained in the working alphabet Γ) in such manner that the initial multisets of the system are multisets over $\Gamma \setminus \Sigma$. Any P system Π with

input membrane has many different *initial configurations*, one for each multiset m over Σ (the multiset m is added to the initial multiset associated with the input membrane). In this case, the system Π with input multiset m is denoted by $\Pi + m$.

Bearing in mind that the solvability of decision problems is defined through the recognition of languages, recognizer membrane systems are introduced in the framework of Membrane Computing.

Definition 1. *A* recognizer membrane system *is a membrane system* (*with or without input membrane*) *such that:* (*a*) *the working alphabet* Γ *contains two distinguished elements* `yes` *and* `no`*;* (*b*) *in the case of a P system without input membrane, the initial multisets are multisets over* Γ*, but in the case of a P system with input membrane, the initial multisets are multisets over* $\Gamma \setminus \Sigma$*, being* Σ *the input alphabet of the system;* (*c*) *all computations halt; and* (*d*) *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 in the output region associated with the corresponding halting configuration of $\mathcal{C}$.

Bearing in mind that every computation in a recognizer membrane system is a halting computation, the left-hand side of any rule of the system must contains at least one object.

These concepts are extended in a natural way to tissue-like P systems inspired by cell inter-communication in tissues. Specifically, if Γ, Σ and $\mathcal{E}$ are the working alphabet, the input alphabet and the alphabet of the environment, respectively, then $\mathcal{E} \subseteq \Gamma \setminus \Sigma$ and the initial multisets of a tissue-like P system are multisets over $\Gamma \setminus \Sigma$.

4 Solving Decision Problems by Families of Membrane Systems

The first results showing that membrane systems could solve computationally hard problems in polynomial time were obtained using P systems without input membrane [6, 21, 33]. This kind of solutions can be considered as *special purpose* solutions: a specific P system is associated with each instance of the problem in such manner that the syntax of the instance is part of the description of the P system.

4.1 Semi-uniform solutions

Next, following [26] the *special purpose* solutions is defined in a mathematical and we will call it *semi-uniform solutions*.

Definition 2. *Let $X = (I_X, \theta_X)$ be a decision problem and let $\mathcal{R}$ be a class of recognizer membrane systems without input membrane. We say that X is* solvable in polynomial time *and* semi-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 $\Pi(u)$ from the instance $u \in I_X$.*
- *The family is polynomially bounded; that is, there exists a natural number $k \in \mathbb{N}$ such that for each instance $u \in I_X$, every computation of $\Pi(u)$ performs at most $|u|^k$ steps.*
- *The family is* sound with respect to X*, that is, for each instance $u \in I_X$, if* there exists *an accepting computation of $\Pi(u)$ then $\theta_X(u) = 1$.*
- *The family is* complete with respect to X*, that is, for each instance $u \in I_X$, if $\theta_X(u) = 1$ then* every *computation of $\Pi(u)$ is an accepting computation.*

According with the previous definition:

– We say that the family $\{\Pi(u) \mid u \in I_X\}$ provides a *semi–uniform solution* to the problem X.
– For each instance $u \in I_X$, the system $\Pi(u)$ processes u. Besides, from the soundness and completeness of the family with respect to the decision problem X it follows that the system $\Pi(u)$ 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.

4.2 Uniform solutions

Next, a new kind of solutions to decision problems by means of families of recognizer membrane systems is introduced. In this context, all instances of the problem with the *same size*, via a given "reasonable encoding scheme", are processed by the same system to which an appropriate input is supplied.

Definition 3. *Let $X = (I_X, \theta_X)$ be a decision problem and let $\mathcal{R}$ be a class of recognizer membrane systems with input membrane. We say that X is* solvable in polynomial time *and* uniform *way by a family $\{\Pi(n) \mid n \in \mathbb{N}\}$ 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 $\{\Pi(n) \mid n \in \mathbb{N}\}$.
– 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 [28].

Obviously, every uniform solution of a decision problem can be considered as 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. It has been proved that the concept uniformity solution is strictly weaker than semi-uniformity solution, for some membrane systems [11].

5 Solving Decision Problems by a Single Membrane System

Definition 4. *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* <u>with</u> *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,* $X \in \mathbf{PMC}^{1f}_{\mathcal{R}}$*, if the following hold:*

- *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 $\mathbf{PMC}^{1f}_{\mathcal{R}} \subseteq \mathbf{PMC}_{\mathcal{R}}$, for every class $\mathcal{R}$ of recognizer membrane systems <u>with</u> input membrane.

6 New Methodology to Tackle the P Versus NP Problem

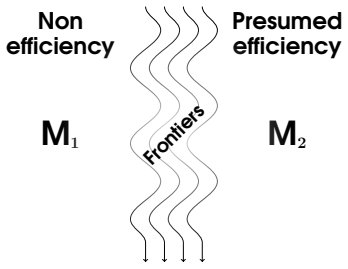
Each computing model provides a mathematical definition of the informal idea of solving abstract problems by means of a mechanical procedure (*algorithm*). A computing model which is equivalent in power to Turing machines is called *universal*.

An abstract problem is said to be *tractable* if it can be solved by a deterministic Turing machine working in polynomial time (the *upper bound* of computational resources is polynomial). The complexity class of tractable decision problems is denoted by **P**. An abstract problem is said to be *intractable* if it cannot be solved by a deterministic Turing machine working in polynomial time (the *lower bound* of the computational resources is exponential).

A computing model with the ability to provide polynomial-time solutions to intractable problems is called *efficient*. In a *non-efficient* computing model, only tractable problems can be solved in polynomial time. It is widely believed that $\mathbf{P} \neq \mathbf{NP}$ and so **NP**-complete problems (the hardest problems in class **NP**) are considered as *presumably intractable* problems. A computing model with the ability to provide polynomial-time solutions to **NP**-complete problems is called *presumably efficient*.

Given two computing models M_1 and M_2 we say that M_1 is a *submodel* of M_2, denoted by $M_1 \subseteq M_2$, if each solution of a problem in M_1 is also a solution in M_2, that is, M_2 is an *extension* of M_1 in the sense that M_2 can be obtained from M_1 by adding some syntactic or semantic ingredients. If M_1 is a non-efficient computing model and M_2 is a presumably efficient one such that $M_1 \subseteq M_2$, then the (syntactical or semantic) ingredients allowing to pass from M_1 to M_2 provide a frontier between the efficiency and the presumed efficiency, that is, passing from M_1 to M_2 amounts to passing from tractability to presumed intractability. Therefore, it gives us a novel tool to tackle the **P** versus **NP** problem as follows:

- In order to show that $\mathbf{P} = \mathbf{NP}$, it is enough to find a polynomial-time solution to one **NP**-complete problem in M_2 and translate it to a polynomial-time solution in M_1, that is, the ingredients added to obtain M_2 from M_1 *do not play a relevant role* in that solution.
- In order to show that $\mathbf{P} \neq \mathbf{NP}$, it is enough to find one **NP**-complete problem that cannot be solved efficiently in M_1, that is, that the ingredients added to obtain M_2 from M_1 are *crucial* to obtain the presumed efficiency.



7 Tractability Frontiers in Membrane Computing

In this section, the methodology to tackle the **P** *versus* **NP** problem, previously described, is applied to the membrane computing paradigm. Specifically, different frontiers between the non efficiency and the presumed efficiency of membrane systems are presented.

In the following, the state of art in terms of computational complexity classes in Membrane Computing is presented in a graphic way. Tractability frontiers can be observed by comparing the nodes of the graphics. Close nodes in the graphs mean computing models relatively similar to each other, either by modifying the length of their rules or by adding or removing some kind of rules.

7.1 P systems with active membranes

P systems with active membranes have a long tradition in computational complexity theory, since they are the first model that was used to provide efficient solutions to hard problems. Figure 1 summarizes most of the results known so far about the power of a number of variants of active membrane models [21, 33, 27, 22, 23, 16, 25, 4].

The different variants of P systems with active membranes mentioned in the figure are the following: $\mathcal{NAM}$ is the class of systems that do not use division rules; $\mathcal{DAM}$ (resp. $\mathcal{SAM}$) is the class of systems using division rules (resp. separation rules); $\mathcal{DAM}^0$ refers to polarizationless P systems from $\mathcal{DAM}$; and $\mathcal{BAM}$ stands for polarizationless P systems using bi-stable catalysts.

More precisely, each node is associated to a complexity class $\mathbf{PMC}_{\mathcal{R}}$ for a particular model of P systems, $\mathcal{R}$, according to the following criterion: each "floor" of this "tower" corresponds to a family of models (as it was just explained), and the four nodes on each level correspond to the variants allowing or forbidding dissolution ($+d$ or $-d$, respectively) and allowing or forbidding division/separation rules for non-elementary membranes ($+ne$ or $-ne$, respectively).

It is interesting to focus on the frontiers of presumed efficiency, that is, edges connecting a non-efficient node ($\mathbf{P} = \mathbf{PMC}_{\mathcal{R}}$) with an efficient or presumably efficient node (having **PSPACE** or **DP** as a lower bound for $\mathbf{PMC}_{\mathcal{R}}$).

7.2 P systems with symport/antiport rules

In this section we display the complexity classes associated with P systems using only two types of rules: communication of symport/antiport type, and rules for increasing the number of membranes (division or separation). It is customary (similarly to the case of tissue-like P systems) to assume that there is an unbounded number of available copies of symbols from the environment alphabet. Figure 2 shows the complexity class associated to each variant with respect to k, that indicates the maximum length allowed for the communication rules [9, 14, 32, 10].

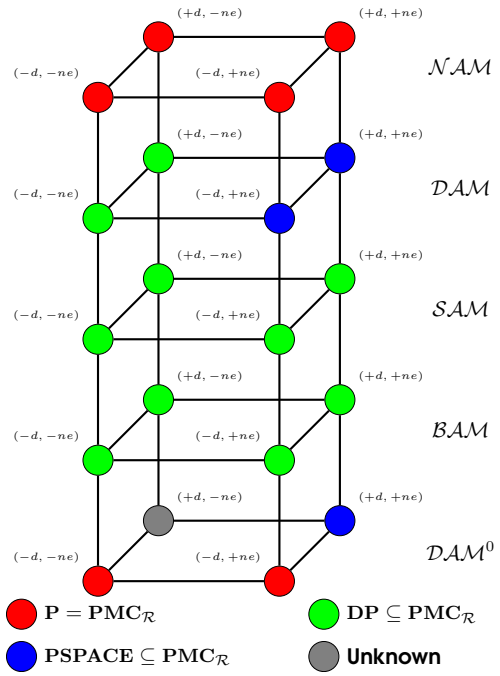


Fig. 1. Borderlines of tractability in the framework of P systems with active membranes

The impact of removing the environment on the power of the corresponding class of P systems is also analyzed [13, 8].

$\mathcal{CDC}$ stands for cell-like P systems using symport/antiport rules as well as division rules; $\widehat{\mathcal{CDC}}$ is the variant of the previous model without environment. Analogously, $\mathcal{CSC}$ and $\widehat{\mathcal{CSC}}$ refer to the case where separation rules are used instead of division. Finally, $\mathcal{CC}$ and $\widehat{\mathcal{CC}}$ stand for the families of P systems using only communication rules.

Note that if neither division nor communication rules are allowed, then the model will never be efficient, irrespectively of the length of the rules, and both with and without environment.

7.3 Tissue P systems with symport/antiport rules

Similar results have been obtained for tissue P systems with symport/antiport rules (cell division/cell separation) and for cell-like P systems with symport/antiport rules (cell division/cell separation). From this perspective, it seems that the underlying structure (*rooted tree* versus *directed graph*) is not relevant.

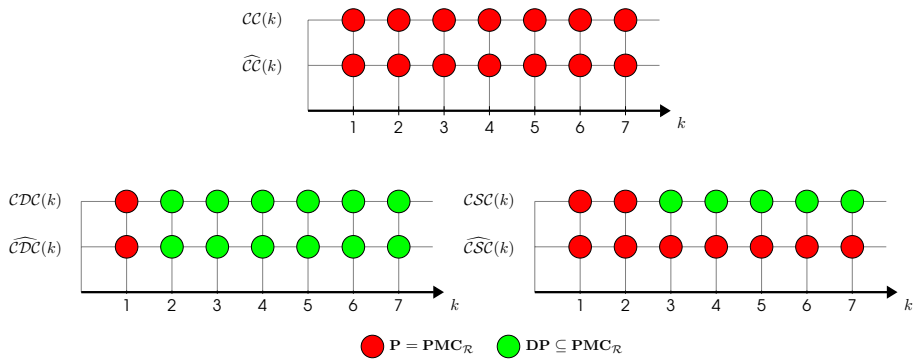


Fig. 2. Borderlines of tractability in the framework of P systems with symport/antiport rules

This can be seen by comparing Figure 2 to Figure 3, where the bounds for complexity classes associated to tissue-like P systems are displayed, again with respect to the length of communication rules [2, 17, 30, 29, 24, 7].

Note that in this case the level of details goes one step further, since it includes the analysis of systems where the type of communication rules is restricted to only symport or only antiport. More precisely, $\mathcal{TDC}$ stands for tissue-like P systems with division rules; $\widehat{\mathcal{TDC}}$ is the variant of the previous model without environment. Analogously, $\mathcal{TSC}$ and $\widehat{\mathcal{TSC}}$ refer to the case where separation rules are used instead of division. If the type of allowed communication rules is restricted to only antiport (resp. only symport), then we can define the corresponding families of P systems $\mathcal{TDA}$ and $\mathcal{TSA}$ (resp. $\mathcal{TDS}$ and $\mathcal{TSS}$). Finally, $\mathcal{TC}$ and $\widehat{\mathcal{TC}}$ stand for the families of tissue-like P systems using only communication rules.

The types of frontiers of presumed efficiency that arise here include: (a) using only communication, or allowing also division or separation (for systems using communication rules of length $k \geq 2$ or $k \geq 3$, respectively); (b) allowing environment or not (for systems using separation rules and communication rules of length $k \geq 3$); (c) choosing between division or separation rules (for systems without environment); and (d) the length of communication rules (moving from $k = 1$ to $k \geq 2$ for systems using division, or moving from $k \leq 2$ to $k \geq 3$ for systems using separation).

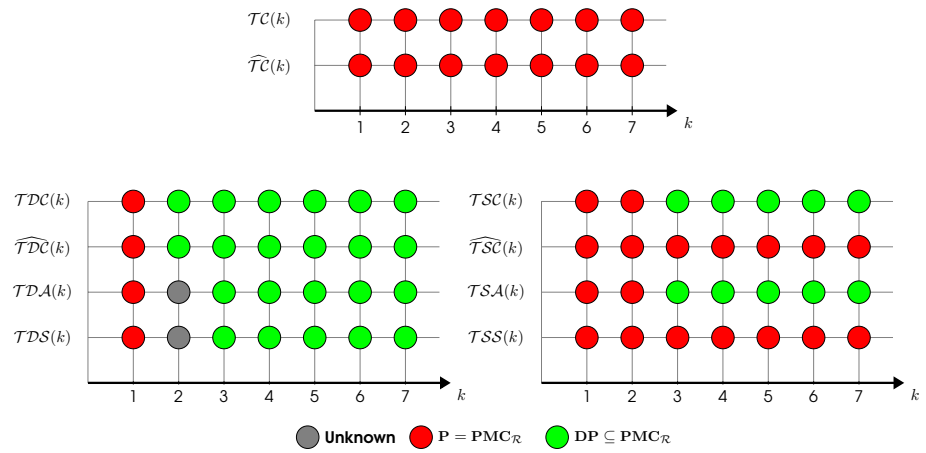


Fig. 3. Borderlines of tractability in the framework of tissue P systems with symport/antiport rules

7.4 Tissue P systems with evolutional symport/antiport rules

In this section, we overview some of the most recent results obtained in the field of computational complexity classes [31, 18, 15, 12]. More precisely, we study tissue P systems using a modified version of symport/antiport rules, where objects on two different regions may cooperate to trigger a rule, and the objects produced by the rule might be placed in any (or both) of these regions.

Figure 4 illustrates the currently known bounds for the complexity classes associated with tissue P systems with evolutional symport/antiport rules, organizing the results according to the total length of communication rules. $\mathcal{TDEC}(k)$ stands for Tissue P systems with evolutional symport/antiport rules of length bounded by k and division rules; $\widehat{\mathcal{TDEC}}(k)$ is the subclass of the previous family where the alphabet of the environment is empty; analogous definitions for $\mathcal{TSEC}(k)$ and $\widehat{\mathcal{TSEC}}(k)$, using separation rules instead of division.

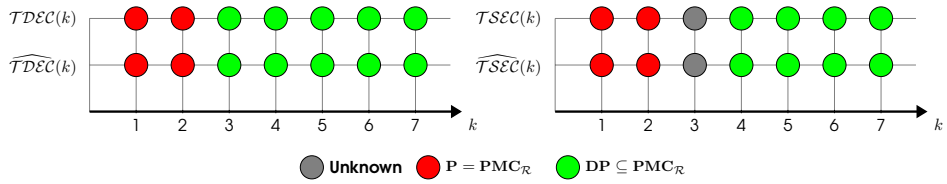


Fig. 4. Borderlines of tractability in the framework of tissue P systems with evolutional symport/antiport rules with respect to the length of the rules.

Since this type of rule seems very powerful, it is natural to try to minimize the cooperation by restricting the length of the rules.

More precisely, in Figure 5, the tractability and presumable efficiency of the complexity classes associated to the corresponding variants of tissue P systems with evolutional symport/antiport rules and division (or separation) rules is illustrated, according to two parameters: length of the left-hand side, k_1, and length of the right-hand side, k_2. Note that the environment does not play a relevant role with respect to the power of these systems.

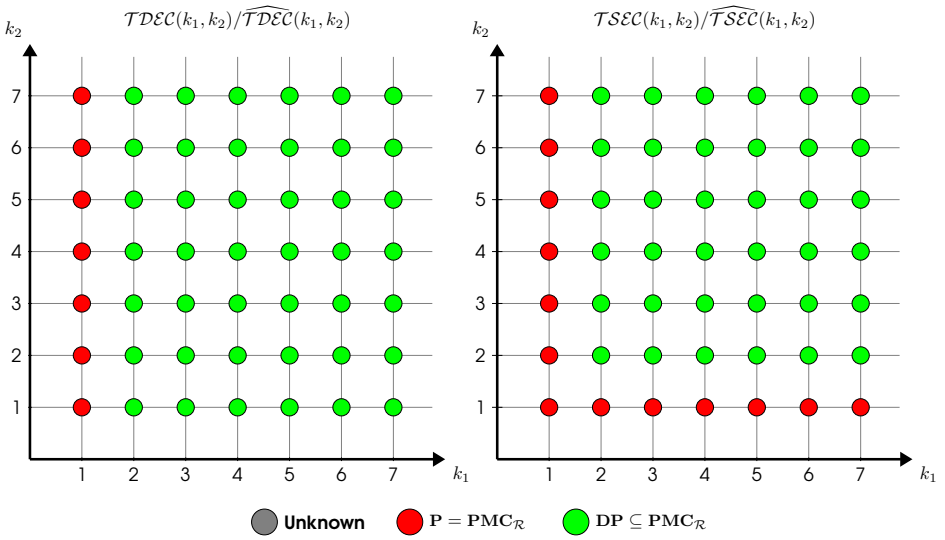


Fig. 5. Borderlines of tractability in the framework of tissue P systems with evolutional symport/antiport rules with respect to the length of the left-hand side, k_1, and length of the right-hand side, k_2.

Acknowledgements

The author is very grateful to Gheorghe Păun for his unconditional support, for his enthusiastic talent to act as a catalyst and for many years of fruitful scientific collaboration and rewarding friendship. Congratulations on the 20^{th} anniversary of the International Conference on Membrane Computing!

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. T.H. Cormen, C.E. Leiserson, R.L. Rivest. *An Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts, 1994.
2. D. Díaz-Pernil, M.J. Pérez-Jiménez, Á. Romero-Jiménez. Efficient simulation of tissue-like P systems by transition cell-like P systems. *Natural Computing*, **8**, 4 (2008), 797.
3. 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.
4. 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, Gr. Rozenberg, A. Salomaa (eds.), *Membrane Computing*. Springer Berlin Heidelberg, 2006. pp. 224–240.
5. 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.
6. 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), 357367.
7. L.F. Macías-Ramos, M.J. Pérez-Jiménez, A. Riscos-Núñez, M. Rius-Font, L. Valencia-Cabrera. The Efficiency of Tissue P Systems with Cell Separation Relies on the Environment. In E. Csuhaj-Varjú, M. Gheorghe, Gr. Rozenberg, A. Salomaa, G. Vaszil (eds.), *Membrane Computing*. Springer Berlin Heidelberg, 2013. pp. 243–256.
8. L.F. Macías-Ramos, M.J. Pérez-Jiménez, A. Riscos-Núñez, L. Valencia-Cabrera. Membrane fission versus cell division: When membrane proliferation is not enough. *Theoretical Computer Science*, **608** (2015), 57–65.
9. L.F. Macías-Ramos, B. Song, T. Song, L. Pan, M.J. Pérez-Jiménez. Limits on Efficient Computation in P Systems with Symport/Antiport Rules. In C. Graciani, Gh. Păun, A. Riscos-Núñez, L. Valencia-Cabrera (eds.), *Proceedings of the Fifteenth Brainstorming Week on Membrane Computing*, Sevilla, Spain, January 31 - February 3 2017. Fénix Editora. pp. 147–160.
10. L.F. Macías-Ramos, B. Song, L. Valencia-Cabrera, L. Pan, M. J. Pérez-Jiménez. Membrane fission: A computational complexity perspective. *Complexity*, **21**, 6 (2016), 321–334.
11. N. Murphy, D. Woods. Uniformity is Weaker than Semi-Uniformity for Some Membrane Systems. *Fundamenta Informaticae*, **134**, 1-2 (2014), 129-152.
12. D. Orellana-Martín. *El problema P versus NP: desarrollo de nuevas técnicas a través de modelos de computación bio-inspirados*. PhD thesis at Universidad de Sevilla, 2018. https://idus.us.es/xmlui/handle/11441/85318
13. D. Orellana-Martín, M. A. Martínez-del Amor, L. Valencia-Cabrera, B. Song, L. Pan, M.J. Pérez-Jiménez. P systems with symport/antiport rules: When do the surroundings matter? *Theoretical Computer Science*, (2018).
14. D. Orellana-Martín, L. Valencia-Cabrera, A. Riscos-Núñez, M. J. Pérez-Jiménez. A path to computational efficiency through membrane computing. *Theoretical Computer Science*, (in press (2019)).
15. D. Orellana-Martín, L. Valencia-Cabrera, B. Song, L. Pan, M.J. Pérez-Jiménez. Tuning frontiers of efficiency in tissue P systems with evolutional communication rules. *Information Sciences*, to be published.

16. L. Pan, T.-O. Ishdorj. P Systems with Active Membranes and Separation Rules. *Journal of Universal Computer Science*, **10**, 5 (2004), 630–649. http://www.jucs.org/jucs_10_5/p_systems_with_active.
17. L. Pan, M.J. Pérez-Jiménez, A. Riscos-Núñez, M. Rius-Font. New frontiers of the efficiency in tissue P systems. In *Pre-Proceedings of the Asian Conference on Membrane Computing (ACMC 2012)*, Huazhong University of Science and Technology, Wuhan, China, October 15-18 2012. pp. 61–73.
18. L. Pan, B. Song, L. Valencia-Cabrera, M.J. Pérez-Jiménez. The computational complexity of tissue P systems with evolutional symport/antiport rules. *Complexity*, **2018**, Article ID 3745210 (2018), 21.
19. C.H. Papadimitriou. *Computational Complexity*, Addison–Wesley, Massachusetts, 1995.
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. 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
22. M.J. Pérez-Jiménez, A. Riscos-Núñez. A Linear-Time Solution to the Knapsack Problem Using P Systems with Active Membranes. In C. Martín-Vide, G. Mauri, Gh. Păun, Gr. Rozenberg, A. Salomaa (eds.), *Membrane Computing*. Springer Berlin Heidelberg, 2004. pp. 250–268.
23. M.J. Pérez-Jiménez, A. Riscos-Núñez. Solving the Subset-Sum problem by P systems with active membranes. *New Generation Computing*, **23**, 4 (2005), 339–356.
24. M.J. Pérez-Jiménez, A. Riscos-Núñez, M. Rius-Font, F.J. Romero-Campero. A polynomial alternative to unbounded environment for tissue P systems with cell division. *Int. J. Comput. Math.*, **90**, 4 (2013), 760–775.
25. M.J. Pérez-Jiménez, F.J. Romero-Campero. Trading Polarization for Bi-stable Catalysts in P Systems with Active Membranes. In G. Mauri, Gh. Păun, M.J. Pérez-Jiménez, Gr. Rozenberg, A. Salomaa (eds.), *Membrane Computing*. Springer Berlin Heidelberg, 2005. pp. 373–388.
26. M.J. Pérez-Jiménez, A. Romero, F. Sancho. 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.
27. M.J. Pérez–Jiménez, A. Romero–Jiménez, F. Sancho–Caparrini: Complexity classes in cellular computing with membranes. *Natural Computing*, **2**, 3 (2003), 265–285.
28. M.J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini. A polynomial complexity class in P systems using membrane division. *Journal of Automata, Languages and Combinatorics*, **11**, 4 (2006), 423-434. A preliminary version in E. Csuhaj-Varj, 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.
29. M.J. Pérez-Jiménez, P. Sosík. An Optimal Frontier of the Efficiency of Tissue P Systems with Cell Separation. *Fundamenta Informaticae*, **138**, 1-2 (2015), 45–60.
30. A.E. Porreca, N. Murphy, M.J. Pérez-Jiménez. An optimal frontier of the efficiency of tissue P systems with cell division. In M. García-Quismondo, L.F. Macías-Ramos,

Gh. Păun, I. Pérez-Hurtado (eds.), *Proceedings of the Tenth Brainstorming Week on Membrane Computing*, vol. II, Sevilla, Spain, January 30 - February 3 2012. Fénix Editora. pp. 141–166.

31. B. Song, C. Zhang, L. Pan. Tissue-like P systems with evolutional symport/antiport rules. *Information Sciences*, **378** (2017), 177–193.
32. L. Valencia-Cabrera, B. Song, L.F. Macías-Ramos, L. Pan, A. Riscos-Núñez, M.J. Pérez-Jiménez. Minimal cooperation in P systems with symport/antiport: A complexity appoach. In L.F. Macías-Ramos, Gh. Păun, A. Riscos-Núñez, L. Valencia-Cabrera (eds.), *Proceedings of the Thirteenth Brainstorming Week on Membrane Computing*, Sevilla, Spain, February 2-6 2015. Fénix Editora. pp. 301–323.
33. C. Zandron, C. Ferretti, G. Mauri. Solving NP-complete problems using P systems with active membranes. In: I. Antoniou, C. S. Calude and M. J. Dinneen (Eds) *Unconventional Models of Computation*, Berlin, Springer, 2000, pp. 289301.
34. G. Zhang, M.J. Pérez-Jiménez, M. Gheorghe. *Real-Life Modelling with Membrane Computing*. Series: Emergence, Complexity and Computation, Volume 25. Springer International Publishing, 2017, X + 367 pages.