CORE Metadata, citation and similar papers at core.ac.uk
Provided by idUS. Depósito de Investigación Universidad de Sevilla

Membrane Computing as a Modelling Tool: Looking Back and Forward from Sevilla

Manuel García-Quismondo, Carmen Graciani and Agustín Riscos-Núñez

Research Group on Natural Computing,
Department of Computer Science and Artificial Intelligence,
Universidad de Sevilla, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
{mgarciaquismondo,cgdiaz,ariscosn}@us.es

Abstract. This paper is a tribute to Prof. Mario de Jesús Pérez-Jiménez. An overview of modelling applications in membrane computing has been compiled, trying to narrate it from a historical perspective and including numerous bibliographical references. Since being exhaustive was obviously out of scope, this quick tour on almost two decades of applications is biased, paying special attention to the contributions in which Prof. Pérez-Jiménez and members of his research group were involved.

1 Introduction

Membrane computing (MC) is a computational discipline that takes inspiration from molecular and cellular biology. Therefore, it seems natural to consider membrane computing as a valuable modelling framework for biological processes. In this sense, several authors considered the possibility to "return some meaningful information to Biology" (see Chap. 1 in [36]) from the very initial stages of this discipline. The majority of the initial research lines in membrane computing were devoted to theoretical results investigating the computational power of different classes of P systems: either proving their Turing completeness or exploring their equivalence with elements of classical formal language hierarchies. However, from the onset the curiosity to work towards applications has been also present. The first book fully devoted to applications in membrane computing appeared in 2006 [19], although applications were carried out since 2000, as we will mention later. For a more up-to-date overview of applications of membrane computing, please visit the *P systems web page* [3] and the *Bulletin of the International Membrane Computing Society* [1]. Interested readers are also advised to refer to two of the most recent volumes: *Applications of Membrane Computing in Systems and Synthetic Biology* [26] and *Real-life Applications with Membrane Computing* [50]. There are also several other overviews of modelling applications, e.g. [30,37,38].

This paper presents a historical (and obviously non-exhaustive) overview of computational modelling approaches within membrane computing, with a special focus on the contributions in which RGNC[1] members were involved.

1.1 Computational Modelling

Computational modelling and simulation are nowadays a cornerstone of the scientific method. Everything starts in Nature, when we identify some physical or biological phenomenon that attracts our interest for some reason. Usually, such intriguing phenomena can be seen as *complex systems*, in the sense that we can describe them as a collection of elements (let us call them "players") that interact following relatively simple rules, exhibiting a complex behaviour of the system as a whole. This behaviour is commonly referred to as emergent, and means that the evolution of the system displays some special properties that are not trivially deduced from the local dynamics of its components.

The first subjective decision to take when designing a model is to determine which are the *relevant* ingredients (players, features, variables) that will compose this model. Judging which ingredients are significant enough to be included is a decision linked to the reason why the phenomenon is of interest. More precisely, Regev and Shapiro explain in [43] that a *good* model should combine the following key desirable properties: relevant, readable, extensible and computationally tractable. Thus, we have to find the balance between two conflicting goals. On the one hand, we should capture as many ingredients as possible to yield a relevant and realistic model. On the other hand, we should try to keep the model as simple as possible, so that it is easy to interpret and work with.

A fundamental stage in the design of a model is validation. To this aim, we need software simulation tools that allow us to run virtual experiments, in order to carry out a reliable analysis of its dynamics under various initial conditions. This is where the concept of practical feasibility comes into play.

2 Historical Overview of Modelling Works in MC

We would like to initiate the narrative of the time line on a very significant event: the Workshop on Multiset Processing that was held in Curtea de Argeş in 2000. This is commonly accepted as the origin of the series of workshops on membrane computing (which later on evolved and gave rise in 2010 to the International Conference on Membrane Computing).

At that time, the limits of the field of membrane computing were still undefined, and the attendants to the meeting came from a variety of backgrounds.

[1] RGNC stands for *Research Group on Natural Computing* from Universidad de Sevilla, also known in the MC community as "Sevilla team".

2.1 In the Beginning, There Was ...

During the above-mentioned workshop, what could be considered as the precursors of future works in the field of modelling in membrane computing were presented.

Banâtre et al. presented a contribution entitled *Gamma and the Chemical Reaction Model: Fifteen Years After* [7]. Gamma was originally proposed in 1986 as a formalism for the definition of concurrent programs. The basic idea underlying the formalism is to describe computation as a form of chemical reaction on a collection of elementary pieces of data. Indeed, the chemical abstract machine paradigm (*CHAM – CHemical Abstract Machine* –, introduced by Berry and Boudol in early 1990s) already includes the notion of "membrane" as a container of elements that react among them.

Nishida presented a contribution entitled *Multiset and K-Subset Transforming Systems* [31], where he provided an example illustrating how to model a chaotic discrete dynamical system by means of a K-subset transforming system. He also provided theoretical results proving the expressive power of this kind of systems. Moreover, one year later, he participated again in the Workshop on Membrane Computing, presenting another application case study: a model for the light reactions of the photosynthesis [32]. There are a couple of observations on this paper: on the one hand, he mentioned a computer simulator as a natural addition to the theoretical model. On the other hand, he suggested the idea to go beyond standard multisets, using multiplicities on any semiring K.

Last, but not least, Suzuki et al. presented a contribution entitled *Artificial Life Applications of a Class of P Systems: Abstract Rewriting Systems on Multisets* [45]. We will refer to this, together with some related works, in the next section.

2.2 The Dawn of Brainstorming Era

The second significant event that we would like to highlight is the Brainstorming Week on Membrane Computing (BWMC). Its first edition was held in Tarragona (Spain) in February 2003.

The Brainstorming Week is a meeting where participants can freely exchange ideas and open problems. It is much more dynamic than a standard workshop or conference, in the sense that the program is proactively set during the gathering and is made up of provocative talks about works in progress that will later on become papers, after fruitful discussions and joint work sessions in a friendly atmosphere. Except for this first edition, the BWMC has been held annually since 2004 in Sevilla (Spain), organised by the RGNC.

In the proceedings volume that was produced after the first meeting, we can find several applications in the form of computational models.

Suzuki and Tanaka revisit in [47] their proposal to use *Abstract rewriting systems on multisets* as a versatile modelling framework able to capture not only dynamics of chemical reactions (e.g. Brusselator model) and population dynamics in a tri-trophic ecological system (both of them already hinted in [45]), but also other applications in medicine such as inflammatory response.

In particular, Suzuki and his collaborators have presented during the Fourth Workshop on Membrane Computing, held later the same year, a model of the p53[2] signalling network [46].

The authors presented a very simple model using a few multiset rewriting rules, allowing symbols to move along two regions: nucleus and cytoplasm. The goal of the paper is to show that the use of P systems as an alternative to traditional rate equation models is useful and practically feasible. The proceedings are available for download via [3], however interested readers may also refer to the chapter "Modeling p53 Signaling Pathways by Using Multiset Processing", in [19].

One can also find in the Brainstorming proceedings an interesting joint work by Ardelean and Besozzi [4], where the authors propose to pay attention to mechano-sensitive channels from a membrane computing perspective.

The idea is further developed in [11,12], where one can find several considerations about the ingredients or features that should be integrated in a P system variant to be considered as a competitive modelling framework. They already mention the idea of adding *variable parameter*s associated with a membrane (in their case related to membrane tension), whose changes are determined by environmental conditions. They point out that this concept is similar to the role of electrical charges in P systems with active membranes [34]. They also acknowledge the advantage of considering rules able to interact with objects at both sides of the membrane, similar to the approach in [9,10]. In particular, for mechano-sensitive channels it is straightforward that the concentration of reactants should be taken into account in the rules, and the authors cite how Suzuki and Tanaka defined their rules as pairs of the type [*condition*, *action*].

Another aspect that should be taken into account in order to capture the dynamics of biological processes is the replacement of maximal parallelism by some alternative semantics that express a probabilistic behaviour, and consequently the dependence on software tools to run multiple simulations [5,6,13].

2.3 Reaching Maturity

In 2003, the Institute for Scientific Information (ISI) listed the seminal paper of membrane computing [35] as *fast breaking paper*. In addition, membrane computing was selected as a *fast emerging area* in computer science. Let us try to keep track of the lines of work related to the modelling of phenomena in membrane computing, which were gradually gaining strength in those first years. Indeed, we can notice an important milestone in 2005, when the collective volume [19] was elaborated. There were several teams working on different approaches, but sharing similar interests.

[2] The p53 protein is closely related to the control of the *apoptosis* process (also known as "programmed cell death"), and therefore is quite relevant in cancer research.

For example, the teams from Universities of Verona and Milano-Bicocca, leaded by Prof. Vincenzo Manca and Prof. Giancarlo Mauri, respectively, have a long tradition in this direction, both in DNA and membrane computing. Their work has produced not only a large number of publications, but also the development of the corresponding software tools and environments for modelling, as well as many PhD Theses: G. Franco *Biomolecular Computing Combinatorial Algorithms and Laboratory Experiments* (2006), L. Bianco *Membrane Models of Biological Systems* (2007), and A. Castellini *Algorithms and Software for Biological MP Modeling by Statistical and Optimization Techniques* (2010).

In 2005, Mario laid the foundation stone of the Sevilla team modelling contributions. More precisely, he designed in collaboration with Romero-Campero (who was at that time one of his PhD students) a model of the Epidermal growth factor receptor (EGFR) signalling network, using *Continuous P systems* [40]. In this framework, objects multiplicity is represented by positive real numbers, and at each instant rules are considered to be applied a positive real number of times. The concept of *computation step* is formally replaced by an *evolution function* working over matrices of real numbers, although in the software simulator it is obviously necessary to work with an approximation. The robustness of this model was studied in [41].

Continuous P Systems

In [40] a continuous variant of P systems was introduced to model the epidermal growth factor receptor (EGFR) signalling cascade. In contrast to the models developed so far, these systems could evolve in every instant by applying a maximal set of rules a positive real number of times each. Another significant difference was that they worked with *continuous multisets*, a mapping from an alphabet Σ to $\mathbb{R}^+$, the set of non-negative real numbers. The rules that are used are of the form

$$\mathcal{R}_l = \{r : u \; [\; v \;]_l \xrightarrow{\mathcal{K}_r} u' \; [\; v' \;]_l\}$$

where u, v, u', v' are standard multisets (i.e. with natural multiplicities) and $\mathcal{K}_r$ is the function that determines the number of times that the rule will be applied, depending on the multiplicities of objects in the current configuration. This variant was inspired by the fact that in vivo chemical reactions evolve in a continuous way following a rate that depends on the concentration of the reactants (mass action law).

The rules, used to model protein-protein interactions taking place in the compartmentalised structure of the living cell, are usually classified as follows:

- Transformation, complex formation and dissociation rules:
  (1) $[\; a \;]_l \rightarrow [\; b \;]_l$
  (2) $[\; a \; b \;]_l \rightarrow [\; c \;]_l$
  (3) $[\; a \;]_l \rightarrow [\; b \; c \;]_l$
- Diffusing in and out:
  (4) $[\; a \;]_l \rightarrow a \; [\;]_l$
  (5) $a \; [\;]_l \rightarrow [\; a \;]_l$

- Binding and unbinding rules:
  (6) $a\ [\ b\ ]_l \rightarrow [\ c\ ]_l$
  (7) $[\ a\ ]_l \rightarrow b\ [\ c\ ]_l$
- Recruitment and releasing rules:
  (8) $a\ [\ b\ ]_l \rightarrow c\ [\ ]_l$
  (9) $c\ [\ ]_l \rightarrow a\ [\ b\ ]_l$

where a, b, c are objects from the working alphabet and l is a membrane label.

An *instantaneous configuration* of a continuous P system Π is a matrix of $\mathcal{A}_{n,m}(\mathbb{R}^+)$ where $a_{i,j}$ (the element in row i and column j) represents the multiplicity of the object c_j in membrane i.

An *evolution* of a continuous P system is a mapping, $E : \mathbb{R}^+ \longrightarrow \mathcal{A}_{n,m}(\mathbb{R}^+)$, which associates each instant $t \in \mathbb{R}^+$ with an instantaneous configuration of the system. The rules are applied during the evolution of the system in a continuous manner according to their rate of application function.

Observe that the effects of the application of the rules are twofold: the multiplicity of objects appearing in the right-hand side of the rules (products) is increased, while at the same time the multiplicity of objects appearing in the left-hand sides (reactants) is decreased.

In order to approximate the evolution of a continuous P systems in a finite set of instants $t_0, \ldots, t_q$ the rectangle rule numerical method to approximate integrals is used. It is supposed that $t_{j+1} - t_j$ for $j = 0, \ldots, q-1$ is fixed, and small enough to assume that all $\mathcal{K}_r$ remains constant in any interval $[t_j, t_{j+1}]$. With this assumption in mind, the number of times that a rule r is applied during one of those intervals is approximately $(t_{j+1} - t_j) \cdot \mathcal{K}_r(E(t_j))$. Therefore, by doing this approximation the evolution of a continuous P system during a time interval $[t_0, t_q]$ is approximated by the computation of a discrete P system that performs q steps working in a bounded parallel manner.

3 At the Crossroads of Cell Biology and Computation

The title of this section was actually the motto of the Seventh WMC, which was celebrated in 2006 in Leiden (The Netherlands).

After previously described work, in [18,33,42], an improved extension of those P systems was presented. This extension considers that the application of rules is not instantaneous, but takes a predefined amount of time.

In literature, each chemical reaction $r : A + B \xrightarrow{c_r} C$ has an associated mesoscopic rate constant c_r. The following type of P systems captures reaction times as waiting times that determine the order in which reactions place. This time will be computed in a deterministic way, for each reaction, in the following way: $\tau_r = \frac{1}{c_r|A||B|}$ where $|A|$ and $|B|$ represent the number of molecules of the two reactants A, B. $\rho_r = c_r|A||B|$ is considered the probability of the rule to be applied in the next step of evolution.

3.1 Deterministic Waiting Times Algorithm

The following algorithm represents the most natural way of defining the evolution of such a P system.

1. Set t as 0.
2. Calculate $WT = \{(\tau_r, r, i)\}$ for all membranes i in the structure μ and for every rule $r \in \mathcal{R}_{l_i}$.
3. Until the time of the simulation t reaches or exceeds a prefixed time T:
   (a) Sort WT according to their waiting time τ.
   (b) Select from WT the tuple with minimal τ, (τ_M, r_M, i_M) (if there are several τ values, then select all).
   (c) Update WT by subtracting τ_M to the τ of all its elements.
   (d) Update t by adding τ_M.
   (e) Apply the selected rules, r_M, in their corresponding membranes, i_M, only once.
   (f) Recalculate the τ only for those rules which are associated with those compartments affected by the applied rules.
   (g) For each of such rules, compare the new τ with its existing τ and update WT as $\min(\tau, WT)$.

This algorithm simulated and translated the signalling cascade of the epidermal growth factor receptor (EGFR). In a similar manner, it was also used over the set of rules describing the Type I and Type II of FAS-induced apoptotic pathway starting with the stimulation of FAS ligand until the activation of the effector Caspase-3 (see [18]).

3.2 Multienvironment P Systems and Multi-compartmental Gillespie Algorithm

Quorum sensing systems in bacteria are fundamental to the control and regulation of cell behaviour. In particular, in order to capture in detail the activation system of a gene regulation system which depends on cell density, the geographical information is very relevant, and this is why it is worth integrating it within the model [8,44].

A *multienvironment P system* is a construct, $ME = (H, \Gamma, G, \mathcal{M}_E, \Pi, k, \mathcal{R}_E)$, where:

1. H is a finite *set of labels* for the environments.
2. $\Gamma = \{o_1, \dots, o_{m_E}\}$ is a finite alphabet of *objects* (also for chemical substances).
3. $G = (V, S)$ is a graph with n_E nodes, V, that represent the environments (labelled with elements from L) and whose edges, S define how the environments are linked.
4. $\mathcal{M}_E = \{(\mathcal{M}_{E_i}, h_i)\}_{1 \le i \le n_E}$ is the *initial configuration* of ME. S associates each environment j in the graph G with a label, $h_i \in H$ and a continuous multiset, $\mathcal{M}_{E_i}$.

5. Π is a continuous P system with $\Gamma \subseteq \Sigma$.
6. $k \in \mathbb{N}$ is the number of copies of the P system Π that are non-deterministically distributed across the different environments in the initial configuration of the system.
7. $\mathcal{R}_E = \{\mathcal{R}_h\}_{h \in H}$ are finite sets of *rules* of one the following forms:
   - $r : (\ u \xrightarrow{\rho_r} u'\ )_h$ where $u, u' \in \Gamma^*$.
   - $c : (\ v\ )_h \xrightarrow{\rho_c} (\ v\ )_{h'}$ where $v \in \Gamma^*$ for the case of movement of different substances from one environment to one of its neighbouring environments.
   - $m : (\ \Pi\ )_h \xrightarrow{\rho_m} (\ \Pi\ )_{h'}$. In addition to the multisets of objects that represent chemical substances, a certain number of copies of P systems are placed inside the environments. These P systems, and all their contents, can move from one environment to another.

Gillespie's algorithm provides an exact method for the stochastic simulation of systems of bio-chemical reactions. The validity of this method has been rigorously proved and successfully used to simulate various biochemical processes. An extension of this algorithm, called Multi-compartmental Gillespie's Algorithm, was introduced in [42]. Unlike the original version, this method considers the existence of multiple disjoint compartments that represent different regions where chemical reactions occur.

Classical Gillespie's Algorithm

Let us consider an enumeration R of all the rules for one of the membranes of the k P systems, including also the environments. Each one of them is considered to be a separate compartment enclosing a volume.

1. Calculate $a_0 = \sum\limits_{r_i \in R} \rho_{r_i}$
2. Generate two random numbers n_1 and n_2 over the unit interval $(0, 1)$
3. Calculate the waiting time $\tau = \frac{1}{a_0} ln(\frac{1}{n_1})$
4. Take the index j of the rule such that $\sum\limits_{1 \leq k \leq j-1} \rho_{r_k} < n_2 \cdot a_0 \leq \sum\limits_{1 \leq k \leq j} \rho_{r_k}$
5. Return (τ, j)

Multi-compartmental Gillespie's Algorithm

Let us consider an enumeration C of all such compartments.

1. Set t as 0.
2. Calculate $WT = \{(\tau, j, i) : i \in C\}$ using Classical Gillespie's algorithm (as described above) to calculate (τ, j) for each compartment.
3. Until the time of the simulation t reaches or exceeds a prefixed time:
   (a) Sort WT according to τ
   (b) Select from WT the tuple with minimal τ, (τ_M, j_M, i_M)
   (c) Update WT subtracting τ_M to each τ
   (d) Apply r_{j_M} in compartment i_M only once.
   (e) For those compartments i affected by the applied rule
         Recalculate their corresponding (τ, j), using again Classical Gillespie's algorithm, and update WT

4 Sevilla's Ark: Giant Pandas, Bearded Vultures and Zebra Mussels

The origin of the fascinating journey of RGNC across the ocean of computational modelling of ecosystems began in 2008. The first case study focused on an ecosystem related to the *Bearded vulture (Gypaetus barbatus)* in the Pyrenees. In this line, *Probabilistic P systems* were presented in [16,17]. Although the results qualitatively agreed with experimental data, this model was intended to be a preliminary proof of concept. Shortly after, the model was extended and improved in [15], by adding more species and features in order to improve the model's accuracy.

The next upgrade on the modelling framework enabled the modelling of geographical information, yielding the so-called *Multienvironment probabilistic functional extended P systems* [14,22]. This is useful to capture, for instance, scavengers moving along different areas looking for food, featuring different environmental conditions for each area, or the expansion of a disease among a population [20]. In parallel with such refinements of the technical details about the syntax and semantics of the type of P systems that were used, there was another evolution going on the software part. In particular, several simulation algorithms were engineered, implementing in different ways the hybridisation between probabilistic rules and their maximally parallel mode of application. An abstract virtual ecosystem having three trophic levels (grass, herbivorous and carnivorous) was designed, to be used as a scalable case study to perform virtual experiments and compare the characteristics of each algorithm [24,25,29]. On the other hand, after several ad-hoc software developments for simulation tools, including end-user GUIs, the RGNC started to work on a general purpose solution: *MeCoSim (Membrane Computing Simulator)* (see [22,39] and visit [2]).

In line with this general purpose long-term approach, a step-by-step protocol for building computational models was presented in [21], using *Population Dynamics P systems* (PDP systems). This is a generalisation of the syntax of previous frameworks, whose integration in *pLinguaCore v3.0* was announced together with a new simulation algorithm in [28]. A noticeable case study is gene networks (dynamics of logical networks [49], or *Arabidopsis thaliana*'s regulating its circadian rhythms [48]).

4.1 Probabilistic Systems

An extension of multienvironment P systems, encompassing them, was proposed in [16,17] with the following changes:

- Each environment contains exactly one P system. In the initial configuration, the multisets associated with each P system is empty.
- Environments and membranes have no associated labels. Previous models usually used exactly one label for environments and distinguished them as different elements in the set of nodes $V = \{e_1, \ldots, e_{n_E}\}$ of the graph G that defines their interconnections. Since the association between membranes and

labels is bijective, the enumeration of membranes sufficiently identifies each membrane. In this enumeration, 0 is reserved for the skin.

- Movements of substances between environments were reduced to just one substance and were generalised to expand its spread capacity, considering that this capacity could also change during the process:
  $c : (\ x\ )_{e_i} \xrightarrow{\rho_c} (\ y_1\ )_{e_{i_1}} \ldots (\ y_h\ )_{e_{i_h}}$
- Polarity was added to the set of properties of each membrane at an instant t. Rules associated with each membrane have the ability to change this polarity:
  $u\ [\ v\ ]_i^{\alpha} \xrightarrow{\rho_{r,e_j}} u'\ [\ v'\ ]_i^{\alpha'}$ with $\alpha \in \{0, +, -\}$.
- Constants associated with rules are changed by computable functions that, given an instant time t, return a real number within the interval $[0, 1]$. For these functions, the following restrictions are imposed:
  - For each environment e_i and object x, the sum of functions associated with the rules from $\mathcal{R}_{e_i}$ whose left-hand side is $(\ x\ )_{e_j}$ coincides with the constant function equal to 1. At each transition step, one of the applicable rules is selected for application according to the "probability" assigned by the functions.
  - For each $u, v \in \Sigma$, $i \in \{1, \ldots, n\}$, $\alpha, \alpha' \in \{0, +, -\}$, the sum of functions associated with the rules from $\mathcal{R}_i$ whose left-hand side is $u\ [\ v\ ]_i^{\alpha}$ and their right-hand side have polarisation α' coincides with the constant function equal to 1. As before, these functions determine which of them is applied.
  - In order to apply several rules to the same membrane simultaneously, all of them must have the same polarity on their right-hand side.
  - If $(\ x\ )_{e_i}$ is the left-hand side of a rule from $\mathcal{R}_{e_i}$ then none of the rules of $\mathcal{R}_0$ has a left-hand side of the form $u\ [\ v\ ]_0^{\alpha}$ for any $u \in \Sigma^*$ that has $x \in u$.
  - The initial configurations for each P system located in each environment and the functions described above may vary between one another.

Binomial Block Based Simulation Algorithm

One of the first simulation algorithms for PDP systems was: ***B***_inomial_ ***B***_lock_ ***B***_ased algorithm_ (BBB) [14]. In this first approach the rules that have exactly the same left sides are organised into a single block. The algorithm consists of a random selection of the blocks, selecting a maximum number of applications for each of them (according to that "common" left side). Then, for each block, a multinomial distribution of the applications of its rules is calculated, according to their probabilities.

Although this simulation algorithm proved to be very useful [14,15], it has some disadvantages as it does not accurately handle the following semantic properties:

- Competition for resources: Rules with partial and not total overlap on their left-hand sides are classified in different blocks, so common objects will not be distributed among them, since selected blocks are executed to the maximum.
- Consistency of rules: It is up to the designer to ensure that there are no inconsistencies.

- The use of probabilistic functions associated with the rules. Only constant probabilities are considered, which will not be the case in future models based on PDP systems.

Direct Non-deterministic Distribution Algorithm with Probabilities (DNDP)

In order to solve these difficulties, two new algorithms have been developed. that accurately capture the dynamics they intend to emulate: the "Direct Non-deterministic Distribution with Probabilities" algorithm (DNDP) [29] and the "Direct distribution based on Consistent Blocks" algorithm (DCBA) [28]. DNDP intends to make a random distribution of rule applications, but this selection process is biased towards the rules that are most likely to be applied. DCBA was conceived to overcome DNDP's accuracy problem by performing a distribution of objects along rule blocks before applying the random distribution process. Although the accuracy achieved by the DCBA is better than that of the DNDP algorithm, the latter is much faster.

In DNDP algorithm the selection is divided into two microphases:

1. A set of consistent applicable rules is calculated. A priori applicable rules (those whose associated probability is greater than 0 in the current configuration) are shuffled. Following this order, a random number of applications is calculated for each rule according to its probability function using a binomial distribution (taking into account, each time, the objects that will be consumed by the applications of the previous rules and that there will be no consistency problems).
2. The multiplicity of some of them is eventually increased to ensure maximum application, thus obtaining a multiset of maximally consistent applicable rules. In order to fairly distribute the objects among the rules, they are iterated in descendant order with respect to the probabilities. Again, each time, one takes into account objects that will be consumed by the applications of the previous rules and consistency problems, but now adding the maximum number of times that they are applicable.

However, the DNDP algorithm still creates some distortion in the distribution of objects between rules with left-hand side overlap. That is, instead of selecting the rules according to their probabilities in a uniform manner, this selection process is biased towards those with the highest probabilities. In addition, the probabilistic distribution of rule executions within blocks will not ultimately follow a multinomial distribution, since competing rules from other blocks may "consume" necessary objects in the selection process.

The DCBA Algorithm

This is where the latest algorithm comes into play. The main idea behind DCBA is to carry out a proportional distribution of objects between consistent blocks of rules (a concept similar, but not identical, to the blocks in BBB as they take into account polarity change), while dealing probabilities.

In this case, the selection stage consists of three phases: Phase 1 distributes objects to the blocks in a certain proportional way, Phase 2 assures the maximality by checking the maximal number of applications of each block, and Phase 3 translates block applications to rule applications by calculating random numbers using the multinomial distribution.

4.2 Probabilistic Guarded P Systems

Probabilistic Guarded P systems [27] can be considered as an evolution of Population Dynamics P Systems specifically oriented for ecological processes. In this context, PGP systems propose a modelling framework for ecology where inconsistency (that is, having two applicable rules such that they cannot be applied simultaneously, because each of them sets a different polarisation on the right side) is managed by the framework itself, instead of delegating to the designer and the simulation algorithms. In addition, by replacing concepts that are foreign to biology (such as electrical polarizations and internal compartment hierarchies) by state variables known as *flags* that are more natural to the experts, thus simplifying communication between expert and designer.

Although PGP systems provide a simplified alternative to PDP systems, some constraints to the supported models are imposed: only models without object competition are allowed.

In order to assist in the definition, analysis, simulation and validation of PDP-based models related to different real-world ecosystems, MeCoSim (a general purpose application to model, design, simulate, analyse and verify different types of models based on P systems), which uses pLinguaCore as its inference engine, has then been used. Also speed-up of the implemented algorithms by using parallel platforms based on GPUs are addressed.

5 Ongoing and Upcoming Modelling Works

We are currently engaged on the research project *Bio-inspired machines on High Performance Computing platforms: a multidisciplinary approach* (TIN2017-89842-P), funded by the Spanish Government. One of the goals is to bridge the gap between HPC platforms architectures and the specifications of a new type of P systems, trying to gain a significant speed-up in simulations. In particular, one of the specific goals is to investigate the invasion of zebra mussel species in Andalusia (along the Guadalquivir river and its surrounding irrigation network), starting from the model which has been already validated for the Ribarroja reservoir [23]. This is a particularly relevant case study due to its ecological and economic impact.

We are also engaged in the research project *Modeling principles of membrane computing models for giant pandas ecosystems*, supported by the National Natural Science Foundation of China (Grant No. 61672437). We are working in collaboration with the Giant Panda Breeding Base, in Chengdu (China), using the controlled environment (in captivity) as a starting point. The most challenging and exciting goal is to eventually extend the model to individuals living in the wild.

Acknowledgement. The authors are very grateful to Mario de Jesús Pérez-Jiménez for his unconditional support, unlimited generosity, patience and enthusiasm, and particularly for his skillful advising and guiding as their "scientific father".

The authors also acknowledge the support from 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. Bulletin of the International Membrane Computing Society. http://membranecomputing.net/IMCSBulletin/
2. MeCoSim Web. http://www.p-lingua.org/mecosim/
3. P Systems Web Page. http://ppage.psystems.eu/
4. Ardelean, I.I., Besozzi, D.: Mechanosensitive channels, a hot topic in (micro)biology: any excitement for P systems? In: Cavaliere, M., Martin-Vide, C., Păun, G. (eds.) Brainstorming Week on Membrane Computing, pp. 32–36 (2003). Rovira i Virgili University, Technical report 26
5. Ardelean, I.I., Cavaliere, M.: Modelling biological processes by using a probabilistic P system software. Nat. Comput. **2**, 173–197 (2003)
6. Ardelean, I.I., Cavaliere, M.: Playing with a probabilistic P system simulator: mathematical and biological problems. In: Cavaliere, M., Martin-Vide, C., Păun, G. (eds.) Brainstorming Week on Membrane Computing, pp. 37–45 (2003). Rovira i Virgili University, Technical report 26
7. Banâtre, J.-P., Fradet, P., Le Métayer, D.: Gamma and the chemical reaction model: fifteen years after. In: Calude, C.S., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2000. LNCS, vol. 2235, pp. 17–44. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45523-X_2
8. Bernardini, F., Gheorghe, M., Krasnogor, N., Muniyandi, R.C., Pérez-Jímenez, M.J., Romero-Campero, F.J.: On P Systems as a modelling tool for biological systems. In: Freund, R., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2005. LNCS, vol. 3850, pp. 114–133. Springer, Heidelberg (2006). https://doi.org/10.1007/11603047_8
9. Bernardini, F., Manca, V.: Dynamical aspects of P systems. Biosystems **70**, 85–93 (2003)
10. Bernardini, F., Manca, V.: P systems with boundary rules. In: Păun, G., Rozenberg, G., Salomaa, A., Zandron, C. (eds.) WMC 2002. LNCS, vol. 2597, pp. 107–118. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36490-0_8
11. Besozzi, D.: Computational and modelling power of P systems. Ph.D. thesis, Universitá degli Studi di Milano-Bicocca, Italy (2004)
12. Besozzi, D., Ardelean, I.I., Mauri, G.: The potential of P systems for modelling the activity of mechanosensitive channels in E. coli. In: Alhazov, A., Martin-Vide, C., Păun, G. (eds.) Workshop on Membrane Computing, WMC-2003, pp. 84–102, 2003. Rovira i Virgili University, Technical report 28

13. Pescini, D., Besozzi, D., Zandron, C., Mauri, G.: Analysis and simulation of dynamics in probabilistic P systems. In: Carbone, A., Pierce, N.A. (eds.) DNA 2005. LNCS, vol. 3892, pp. 236–247. Springer, Heidelberg (2006). https://doi.org/10.1007/11753681_19
14. Cardona, M., et al.: A computational modeling for real ecosystems based on P systems. Nat. Comput. **10**, 39–53 (2011)
15. Cardona, M., Colomer, M.A., Margalida, A., Pérez-Hurtado, I., Pérez-Jiménez, M.J., Sanuy, D.: A P system based model of an ecosystem of some scavenger birds. In: Păun, G., Pérez-Jiménez, M.J., Riscos-Núñez, A., Rozenberg, G., Salomaa, A. (eds.) WMC 2009. LNCS, vol. 5957, pp. 182–195. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11467-0_14
16. Cardona, M., Colomer, M.A., Pérez-Jiménez, M.J., Sanuy, D., Margalida, A.: A P system modeling an ecosystem related to the bearded vulture. In: 6th Brainstorming Week on Membrane Computing, vol. 6, pp. 51–66 (2008)
17. Cardona, M., Colomer, M.A., Pérez-Jiménez, M.J., Sanuy, D., Margalida, A.: Modeling ecosystems using P systems: the bearded vulture, a case study. In: Corne, D.W., Frisco, P., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2008. LNCS, vol. 5391, pp. 137–156. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-95885-7_11
18. Cheruku, S., Păun, A., Romero-Campero, F.J., Pérez-Jiménez, M.J., Ibarra, O.: Simulating FAS-induced apoptosis by using P systems. Prog. Nat. Sci. **17**, 424–431 (2007)
19. Ciobanu, G., Pérez-Jiménez, M.J., Păun, G. (eds.): Applications of Membrane Computing. Springer, Berlín (2006). https://doi.org/10.1007/3-540-29937-8
20. Colomer, M.A., et al.: Modeling population growth of Pyrenean Chamois (Rupicapra p. pyrenaica) by using P-systems. In: Gheorghe, M., Hinze, T., Păun, G., Rozenberg, G., Salomaa, A. (eds.) CMC 2010. LNCS, vol. 6501, pp. 144–159. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-18123-8_13
21. Colomer, M.A., Margalida, A., Pérez-Jiménez, M.J.: Population dynamics P system (PDP) models: a standardized protocol for describing and applying novel bio-inspired computing tools. PLOS One **4**, 1–13 (2013)
22. Colomer, M.A., Margalida, A., Sanuy, D., Pérez-Jiménez, M.J.: A bio-inspired computing model as a new tool for modeling ecosystems: the avian scavengers as a case study. Ecol. Model. **222**, 33–47 (2011)
23. Colomer, M.A., Margalida, A., Valencia-Cabrera, L., Palau, A.: Application of a computational model for complex fluvial ecosystems: the population dynamics of zebra mussel Dreissena polymorpha as a case study. Ecol. Complex. **20**, 116–126 (2014)
24. Colomer, M.A., Pérez-Hurtado, I., Pérez-Jiménez, M.J., Riscos-Núñez, A.: Simulating tritrophic interactions by means of P systems. In: Nagar, A., Thamburaj, R., Li, K., Tang, Z., Li, R. (eds.) IEEE Fifth International Conference on Bio-inpired Computing: Theories and Applications (BIC-TA 2010), Liverpool, UK, vol. 2, pp. 1621–1628. IEEE Press, September 2010
25. Colomer, M.A., Pérez-Hurtado, I., Riscos-Núñez, A., Pérez-Jiménez, M.J.: Comparing simulation algorithms for multienvironment probabilistic P system over a standard virtual ecosystem. Nat. Comput. **11**, 369–379 (2011)
26. Frisco, P., Gheorghe, M., Pérez-Jiménez, M.J. (eds.): Applications of Membrane Computing in Systems and Synthetic Biology. ECC, vol. 7. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-03191-0

27. García-Quismondo, M., Martínez-del Amor, M.A., Pérez-Jiménez, M.J.: Probabilistic guarded P systems, a formal definition. In: Twelfth Brainstorming Week on Membrane Computing (BWMC2014), pp. 183–206 (2014)
28. Martínez-del Amor, M.A., et al.: DCBA: simulating population dynamics P systems with proportional object distribution. In: Tenth Brainstorming Week on Membrane Computing, vol. II, pp. 27–56, February 2012
29. Martínez-del Amor, M.A., Pérez-Hurtado, I., Pérez-Jiménez, M.J., Riscos-Núñez, A., Colomer, M.A.: A new simulation algorithm for multienvironment probabilistic P systems. In: Li, K., Tang, Z., Li, R., Nagar, A., Thamburaj, R. (eds.) IEEE Fifth International Conference on Bio-inpired Computing: Theories and Applications (BIC-TA 2010), Changsha, China, vol. 1, pp. 59–68. IEEE Press, September 2010
30. Milazzo, P., Pérez-Jiménez, M.J.: Proceedings First Workshop on Applications of Membrane Computing, Concurrency and Agent-Based Modelling in Population Biology. Electronic Proceedings in Theoretical Computer Science, vol. 33 (2010)
31. Yasunobu Nishida, T.: Multiset and K-subset transforming systems. In: Calude, C.S., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2000. LNCS, vol. 2235, pp. 255–265. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45523-X_13
32. Nishida, T.Y.: Simulations of photosynthesis by a k-subset transforming system with membrane. Fundam. Inform. **49**, 249–259 (2002)
33. Păun, A., Pérez-Jiménez, M.J., Romero-Campero, F.J.: Modeling signal transduction using P systems. In: Hoogeboom, H.J., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2006. LNCS, vol. 4361, pp. 100–122. Springer, Heidelberg (2006). https://doi.org/10.1007/11963516_7
34. Păun, G.: P systems with active membranes: attacking NP-complete problems. CDMTCS Research Report Series (1999)
35. Păun, G.: Computing with membranes. J. Comput. Syst. Sci. **61**, 108–143 (2000). Prelim. Version in Turku Center for Computer Science-TUCS Report No. 208 (1998). www.tucs.fi
36. Păun, G.: Membrane Computing. An Introduction. Springer, Berlin (2002). https://doi.org/10.1007/978-3-642-56196-2
37. Păun, G., Pérez-Jiménez, M.J.: Membrane computing: brief introduction, recent results and applications. BioSystems **85**, 11–22 (2006)
38. Păun, G., Romero-Campero, F.J.: Membrane computing as a modeling framework. Cellular systems case studies. In: Bernardo, M., Degano, P., Zavattaro, G. (eds.) SFM 2008. LNCS, vol. 5016, pp. 168–214. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68894-5_6
39. Pérez-Hurtado, I., Valencia-Cabrera, L., Pérez-Jiménez, M.J., Colomer, M.A., Riscos-Núñez, A.: MeCoSim: a general purpose software tool for simulating biological phenomena by means of P systems. In: Li, K., Tang, Z., Li, R., Nagar, A., Thamburaj, R. (eds.) IEEE Fifth International Conference on Bio-inpired Computing: Theories and Applications (BIC-TA 2010), vol. I, pp. 637–643. IEEE Inc., Changsha (2010)
40. Pérez-Jiménez, M.J., Romero-Campero, F.J.: Modelling EGFR signalling network using continuous membrane systems. In: Third International Workshop on Computational Methods in Systems Biology, CMSB 2005, pp. 118–129, 3–5 April 2005
41. Pérez-Jiménez, M.J., Romero-Campero, F.J.: A study of the robustness of the EGFR signalling cascade using continuous membrane systems. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2005. LNCS, vol. 3561, pp. 268–278. Springer, Heidelberg (2005). https://doi.org/10.1007/11499220_28

42. Pérez-Jiménez, M.J., Romero-Campero, F.J.: P systems, a new computational modelling tool for systems biology. In: Priami, C., Plotkin, G. (eds.) Transactions on Computational Systems Biology VI. LNCS, vol. 4220, pp. 176–197. Springer, Heidelberg (2006). https://doi.org/10.1007/11880646_8
43. Regev, A., Shapiro, E.: Cellular abstractions: cells as computations. Nature **419**(6905), 343 (2002)
44. Romero-Campero, F.J., Pérez-Jiménez, M.J.: A model of the quorum sensing system in Vibrio fischeri using P systems. Artif. Life **14**, 95–109 (2008)
45. Suzuki, Y., Fujiwara, Y., Takabayashi, J., Tanaka, H.: Artificial life applications of a class of P systems: abstract rewriting systems on multisets. In: Calude, C.S., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2000. LNCS, vol. 2235, pp. 299–346. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45523-X_16
46. Suzuki, Y., Ogishima, S., Tanaka, H.: Modeling the p53 signaling network by using P systems. In: Alhazov, A., Martin-Vide, C., Păun, G. (eds.) Workshop on Membrane Computing, WMC-2003, pp. 449–454, 2003. Rovira i Virgili University, Technical report 28
47. Suzuki, Y., Tanaka, H.: Abstract rewriting systems on multisets and their application for modelling complex behaviours. In: Cavaliere, M., Martin-Vide, C., Păun, G. (eds.) Brainstorming Week on Membrane Computing, pp. 313–331 (2003). Rovira i Virgili University, Technical report 26
48. Valencia-Cabrera, L., García-Quismondo, M., Pérez-Jiménez, M.J., Su, Y., Yu, H., Pan, L.: Analysing gene networks with PDP systems. Arabidopsis thaliana, a case study. In: Eleventh Brainstorming Week on Membrane Computing (11 BWMC), pp. 257–272, August 2013
49. Valencia-Cabrera, L., García-Quismondo, M., Pérez-Jiménez, M.J., Su, Y., Yu, H., Pan, L.: Modeling logic gene networks by means of probabilistic dynamic P systems. Int. J. Unconv. Comput. **9**(5–6), 445–464 (2013)
50. Zhang, G., Pérez-Jiménez, M.J., Gheorghe, M.: Real-Life Applications with Membrane Computing. ECC, vol. 25. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-55989-6