Comparing simulation algorithms for multienvironment probabilistic P systems over a standard virtual ecosystem

M. Angels Colomer
Ignacio Pérez-Hurtado
Mario J. Pérez-Jiménez
Agustín Riscos-Núñez

Abstract Membrane Computing has recently proved to be a suitable framework for addressing the modelling of dynamical biological systems in general, and ecosystems in particular. Due to the inherent randomness and uncertainty in biological systems, when designing a model the relevant tasks to be addressed are the validation and virtual experimentation processes, rather than the formal verification. It is therefore crucial to rely on software implementations of efficient simulation algorithms. This paper presents a simple (but realistic enough) ecosystem where a carnivore and several herbivorous species interact. The model of this ecosystem has been used to compare experimentally the performance of two different simulation algorithms.

Keywords Ecosystem modelling · Membrane computing · Probabilistic simulation algorithms

1 Introduction

Designing a model for a biological system is an intrinsically complicated task, since there are usually a large number of important factors that need to be considered. It is therefore advisable to make efforts to minimize the number of variables (and, of course, the number of interactions between them).

Nowadays ordinary differential equations (ODEs) constitute the most widely used approach for the study of population dynamics, but this approach has some drawbacks. On one hand, when the number of species in a model is greater than two, the equations system proposed is so complex that it is usually solved using numerical methods. Besides, slight modifications on the modeled population imply important variations on the formal model (the whole modeling process needs to be done again from scratch).

There already exists a quite large literature concerning several approaches to modeling different phenomena within the membrane computing framework (see e.g. Besozzi et al. 2008; Pérez-Jiménez and Romero 2006; Romero and Pérez-Jiménez 2008 and the chapter devoted to *Probabilistic/stochastic models* in the handbook; Păun et al. 2010). Computing models based on P systems offer significant advantages: modularity, parallelism, and no limitation on the number of interrelated variables that evolve in parallel. These properties make them very attractive for modeling complex ecosystems.

P systems explicitly represent the discrete character of the quantity of components of a cellular system by using rewriting rules on multisets of objects which represent molecules, and strings which describe the organisation of genes.

Each ecosystem has its own important peculiarities, but nevertheless, there are some aspects common to most ecosystems such as:

– they contain a large number of individuals and a large number of species.

M. A. Colomer
Department of Mathematics, University of Lleida, Av. Alcalde Rovira Roure, 191, 25198 Lleida, Spain
e-mail: colomer@matematica.udl.cat

I. Pérez-Hurtado · M. J. Pérez-Jiménez · A. Riscos-Núñez (✉)
Research Group on Natural Computing, Department of Computer Science and Artificial Intelligence, University of Sevilla, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
e-mail: ariscosn@us.es

I. Pérez-Hurtado
e-mail: perezh@us.es

M. J. Pérez-Jiménez
e-mail: marper@us.es

– the life cycle includes some basic processes such as: feeding, growth, reproduction and death.
– these processes are annually repeated.
– the evolution often depends on the environment: climate, soil, etc.
– the natural dynamics suffer modifications due to human activities.

These common features yield some requisites for the model, from a computational point of view: many processes take place simultaneously, there is cooperation between individuals and elements of the ecosystem, partial synchronization among the dynamic evolution sub-ecosystems (for example, there could be adverse weather conditions some year, and this does not affect a single sub-ecosystem, but has a global influence on the entire ecosystem), situations need to be restored annually.

These considerations led to the definition of an appropriate modeling semantic context for the P system. In particular, a precise semantics of the multienvironment functional probabilistic P system with active membranes has been used to model two real ecosystems: One, dealing with the scavenger birds in the Catalan Pyrenees (Spain) (Cardona et al. 2010) and another one focusing on the *zebra mussel* in the reservoir of Ribarroja (Spain) (Cardona et al. 2011). In the first case, the purpose of the obtained model is the study of the evolution of the ecosystem under different scenarios to make the most appropriate management decisions for the conservation of an endangered species. The second case study corresponds to a completely different situation: zebra mussel is an exotic species that has shown an excellent adaptation after being introduced in the reservoir. Its uncontrolled reproduction causes significant economic and ecological damage. Hence, the goal in this case is to learn how to reduce to the limit of the mussel population.

In both cases we have designed a simulator to validate the models. Actually, two different tools have been handled to the corresponding managers, enabling them to perform virtual experiments under different conditions.

This paper introduces a model for an idealized ecosystem that can be used when testing simulators, instead of running tests on huge instances like the models corresponding to the above mentioned real case studies. The presented ecosystem contains species belonging to three trophic levels: in the lower level we have the grass, then we have 5 herbivore species, and above all of them we have a carnivore at the third level. Although it does not correspond to any real system, it has been designed under the guidance of ecologists, and it takes into account some relevant facts that make it somehow plausible. Some of these details will be explained later on.

The paper is structured as follows. The next section describes in general the modeling framework based on P systems. In Sect. 3 the main contribution of the paper is detailed, according to the previous framework. Then, two different simulation algorithms are explained, and simulation results comparing their performance are shown in Sect. 6 The paper ends with some conclusions and final remarks.

2 A P system based modeling framework

In this section, a model with a network of *environments*, each of them containing a P system, is presented. All P systems share the same *skeleton*, in the sense that they have the same working alphabet, the same membrane structure and the same set of rules. However, as it will be explained in the following definition, the probability to apply a rule can vary for each environment.

Definition 1 A multienvironment probabilistic P system of degree (q, m) with $q \geq 1$, $m \geq 1$, taking T time units, $T \geq 1$, is a tuple $(G, \Gamma, \Sigma, T, R_E, \mu, R, \{f_{r,j} : r \in R, 1 \leq j \leq m\}, \{\mathcal{M}_{ij} : 0 \leq i \leq q-1, 1 \leq j \leq m\})$ where:

– $G = (V, S)$ is a directed graph. Let $V = \{e_1, \ldots, e_m\}$ whose elements are called environments;
– Γ is the working alphabet and $\Sigma \subsetneq \Gamma$ is an alphabet representing the objects that can be present in the environments;
– T is a natural number that represents the simulation time of the system;
– R_E is a finite set of communication rules between environments of the form

$$(x)_{e_j} \xrightarrow{p_{(x,j,j_1,\ldots,j_h)}} (y_1)_{e_{j_1}} \ldots (y_h)_{e_{j_h}}$$

where $x, y_1, \ldots, y_h \in \Sigma, (e_j, e_{j_l}) \in S (l = 1, \ldots, h)$ and $p_{(x,j,j_1,\ldots,j_h)}(t) \in [0, 1]$, for each $t = 1, \ldots, T$. If $p_{(x,j,j_1,\ldots,j_h)}(t) = 1$, for each t, then we omit the probabilistic function. These rules verify the following:

- For each environment e_j and for each object x, the sum of functions associated with the rules from R_E whose left-hand side is $(x)_{e_j}$ coincides with the constant function equal to 1.

– μ is a membrane structure consisting of q membranes, with the membranes injectively labeled by $0, \ldots, q-1$. The skin membrane is labeled by 0. We also associate electrical charges from the set $\{0, +, -\}$ with membranes.
– R is a finite set of evolution rules of the form $r : u[v]_i^{\alpha} \rightarrow u'[v']_i^{\alpha'}$ where u, v, u', v' are multisets over $\Gamma, i \in \{0, 1, \ldots, q-1\}$, and $\alpha, \alpha' \in \{0, +, -\}$.
– For each $r \in R$ and for each j, $1 \leq j \leq m$, $f_{r,j}$ is a computable function whose domain is $\{1, \ldots, T\}$ and its range is contained in [0, 1], verifying the following:

- For each $u, v \in \Gamma^*, i \in \{0, \ldots, q-1\}$ and $\alpha, \alpha' \in \{0, +, -\}$, if $r_1, \ldots, r_z$ are the rules from R whose left-hand side is $u[v]_i^\alpha$ and the right-hand side have polarization α', then $\sum_{j=1}^{z} f_{r_j}(t) = 1$, for each t, $1 \leq t \leq T$.
- If $(x)_{e_j}$ is the left-hand side of a rule $r \in R_E$, then none of the rules of R has a left-hand side of the form $u\ [\ v\ ]_0^\alpha$, for any $u, v \in \Gamma^*$ and $\alpha \in \{0, +, -\}$, having $x \in u$.

– For each $j (1 \leq j \leq m)$, $\mathcal{M}_{0j}, \ldots, \mathcal{M}_{q-1,j}$ are strings over Γ, describing the multisets of objects initially placed in the q regions of μ, within the environment e_j.

In other words, a system as described in the previous definition can be viewed as a set of m environments $e_1, \ldots, e_m$ linked between them by the arcs from the directed graph G. Each environment e_j contains a P system, $\Pi_j = (\Gamma, \mu, R, \mathcal{M}_{0j}, \ldots, \mathcal{M}_{q-1,j})$, of degree q, such that $\mathcal{M}_{0j}, \ldots, \mathcal{M}_{q-1,j}$ describe the initial multisets for this environment, and every rule $r \in R$ has a computable function $f_{r,j}$ (specific for environment j) associated with it.

The tuple of multisets of objects present at any moment in the m environments and at each of the regions of each Π_j, together with the polarizations of the membranes in each P system, constitutes a *configuration* of the system at that moment. At the initial configuration of the system we assume that all environments are empty and all membranes have a neutral polarization.

We assume that a global clock exists, marking the time for the whole system, that is, all membranes and the application of all rules (both from R_E and R) are synchronized in all environments.

The P system can pass from one configuration to another by using the rules from $R = R_E \cup \bigcup_{j=1}^{m} R_{\Pi_j}$ as follows: at each transition step, the rules to be applied are selected according to the probabilities assigned to them, and all applicable rules are simultaneously applied.

When a communication rule between environments

$$(x)_{e_j} \xrightarrow{p_{(x,j,j_1,\ldots,j_h)}} (y_1)_{e_{j_1}} \ldots (y_h)_{e_{j_h}}$$

is applied, object x passes from e_j to $e_{j_1}, \ldots, e_{j_h}$ possibly modified into objects $y_1, \ldots, y_h$, respectively. At any moment t, $1 \leq t \leq T$, for each object x in environment e_j, if there exist communication rules whose left-hand side is $(x)_{e_j}$, then one of these rules will be applied. If more than one communication rule can be applied to an object, the system selects one randomly, according to their probability which is given by $p_{(x,j,j_1,\ldots,j_h)}(t)$.

For each j $(1 \leq j \leq m)$ there is just one further restriction, concerning the consistency of charges: in order to apply several rules of R_{Π_j} simultaneously to the same membrane, all the rules must have the same electrical charge on their right-hand side.

3 A P system based model of tritrophic interactions

Feeding, reproduction and mortality are basic processes for all living organisms. These natural processes are those considered in the model of tritrophic interactions presented here. The model is a simplification of a real ecosystem and many details have been skipped. It should be considered as a simple approximation facilitating a better understanding of the methodology that allows to model actual ecosystems by means of P systems (see e.g. Cardona et al. 2010, 2011 where two existing ecosystems of great ecological interest are modeled).

The model scheme is composed by 5 modules as shown in Fig. 1. A complete cycle is executed in 14 steps of computation and it represents one year in the ecosystem. We will divide the ecosystem into 10 areas with different weather, orography, and soil conditions, each of them having an environment associated with it. The system includes an additional environment with no geographical meaning, e_{11}, that will be used to control and synchronize the application of the modules.

The ecosystem is modeled by using a multienvironment probabilistic P system of degree (11,2) taking T time units:

$$(G, \Gamma, \Sigma, T, R_E, \mu, R, \{f_{r,j} : r \in R, 1 \leq j \leq 11\}, \{\mathcal{M}_{0j}, \mathcal{M}_{1j} : 1 \leq j \leq 11\})$$

where:

– The graph of the system is $G = (V, S)$, where $V = \{e_1, \ldots, e_{11}\}$ and $S = \{(e_i, e_{11}), (e_{11}, e_i)) : 1 \leq i \leq 10\}$.
– $\Gamma = \Sigma \cup \{X_1\} \cup \{V_i : 2 \leq i \leq 7\} \cup \{Z_i : 0 \leq i \leq 11\}$.
$\Sigma = \{G\} \cup \{G'_k : 1 \leq k \leq 10\} \cup \{X_i, Y_i, W_i : 2 \leq i \leq 7\}$
$\cup \{X'_{i,k}, Y'_{i,k}, W'_{i,k} : 2 \leq i \leq 7, 1 \leq k \leq 10\}$.

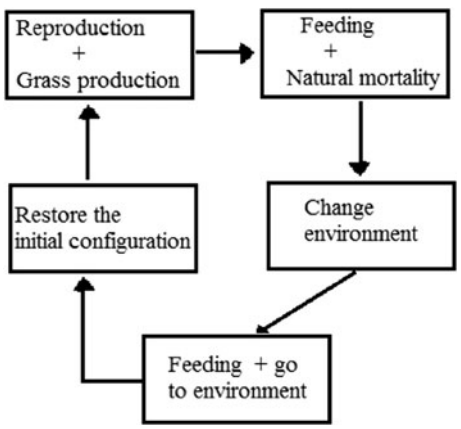


Fig. 1 Modules scheme

The symbols X_i, Y_i, V_i, W_i, $X'_{i,k}$ and $Y'_{i,k}$, with $2 \leq i \leq 7$ and $1 \leq k \leq 10$ represent animals of species i and environment k. X_1 is associated with 1 hectare (He) of pasture. The symbols G and G' represent 1 kg of grass and Z_i with $0 \leq i \leq 11$ is an auxiliary synchronization counter.

- $\Pi = (\Gamma, [[\,]_1]_0, R_\Pi)$ is the common skeleton for all environments.
- R_E is the set of rules for communication between environments, and they will be described later on (within Module 3 and Module 5).
- $\{f_{r,j} : r \in R_\Pi, 1 \leq j \leq 10\}$ is a set of constant functions whose range is contained in [0, 1].
- $\{\mathcal{M}_{0j}, \mathcal{M}_{1j} : 1 \leq j \leq 10\}$ are strings over Γ, describing the multisets of objects initially placed in the two regions of μ for each $\Pi_j : 1 \leq i \leq 10$.
  - $\mathcal{M}_{0j} = Z_0 X_1^{q_{1,j}}$, for $1 \leq j \leq 10$.
  - $\mathcal{M}_{1j} = X_2^{q_{2,j}} \ldots X_7^{q_{7,j}}$, for $1 \leq j \leq 10$.
  - $\mathcal{M}_{0,11} = Z_0$, and $\mathcal{M}_{1,11} = \lambda$.
- $R = R_E \cup \bigcup_{j=1}^{m} R_{\Pi_j}$ is a set of rules composed by the rules described below:

In order to synchronize the P system we use the object Z_i, the evolution of this object is made according to the rules

$r_1 \equiv Z_0[\,]_1^0 \to [Z_0]_1^0$
$r_2 \equiv Z_0[\,]_1^0 \to [Z_1]_1^-$.
$r_3 \equiv Z_1[\,]_1^- \to [Z_2]_1^0$.
$r_4 \equiv Z_n[\,]_1^- \to [Z_{n+1}]_1^0, 1 \leq n \leq 10, n \neq 6, 7$.
$r_5 \equiv Z_6[\,]_1^0 \to [Z_7]_1^-$.
$r_6 \equiv Z_7[\,]_1^- \to [Z_8]_1^0$.
$r_7 \equiv Z_{11}[\,]_1^0 \to [Z_0]_1^+$.
$r_8 \equiv [Z_0]_1^+ \to Z_0[\,]_1^0$.

- **Module 1** (Grass production and species reproduction):
  - Grass production.
    $r_9 \equiv X_1[\,]_1^0 \xrightarrow{m_j} [X_1 G^{h_j}]_1^0, 1 \leq j \leq 3$.
    The environmental conditions affect the amount of produced grass per unit of surface (He). For the sake of simplicity only three scenarios are considered: low, normal and high production (with probabilities m_1, m_2 and m_3, respectively).
  - Females which reproduce and generate d_i descendants.
    $r_{10} \equiv [X_i]_1^0 \xrightarrow{k_{i,1} \cdot 0.5} [V_i^{1+d_i}]_1^0, 2 \leq i \leq 7$.
  - Females and males which do not reproduce.
    $r_{11} \equiv [X_i]_1^0 \xrightarrow{1-k_{i,1} \cdot 0.5} [V_i]_1^0, 2 \leq i \leq 7$.
    A 1:1 ratio between females and males is assumed. Moreover, all the females are able to reproduce. The first of these assumptions is fulfilled by most of animal species, but not the second one, since fertility depends on the age.
- **Module 2** (Feeding and mortality):
  - Animals which feed and survive.
    $r_{12} \equiv [V_i G^{f_i}]_1^0 \xrightarrow{1-k_{i,2}} [Y_i]_1^-, 2 \leq i \leq 6$.
    $r_{13} \equiv [V_7 V_i^{f_7}]_1^+ \xrightarrow{1-k_{7,2}} [Y_7]_1^-, 2 \leq i \leq 6$.
  - Animals which feed and do not survive.
    $r_{14} \equiv [V_i G^{f_i}]_1^0 \xrightarrow{k_{i,2}} [\,]_1^-, 2 \leq i \leq 6$.
    $r_{15} \equiv [V_7 V_i^{f_7}]_1^0 \xrightarrow{k_{7,2}} [\,]_1^-, 2 \leq i \leq 6$.
- **Module 3** (Communication between environments):
  - All animals should go to the environment e_{11}, so the first step is to go to the skin membrane.
    $r_{16} \equiv [V_i]_1^- \to X_i[\,]_1^0, 2 \leq i \leq 7$.
    $r_{17} \equiv [Y_i]_1^- \to W_i[\,]_1^0, 2 \leq i \leq 6$.
    $r_{18} \equiv [G]_1^- \to G[\,]_1^0$.
  - The objects associated to the food and animals that are in the skin membrane should go to the environment.
    $r_{19} \equiv [X_i]_0^0 \to X_i[\,]_0^0, 2 \leq i \leq 7$.
    $r_{20} \equiv [W_i]_0^0 \to W_i[\,]_0^0, 2 \leq i \leq 6$.
    $r_{21} \equiv [G]_0^0 \to G[\,]_0^0$.
  - The objects that are in the environments associated with the geographical areas go to the virtual environment e_{11}.
    $r_{22} \equiv (X_i)_{e_k} \to \left(X'_{i,k}\right)_{e_{11}}, 2 \leq i \leq 7, 1 \leq k \leq 10$.
    $r_{23} \equiv (W_i)_{e_k} \to \left(W'_{i,k}\right)_{e_{11}}, 2 \leq i \leq 7, 1 \leq k \leq 10$.
    $r_{24} \equiv (G)_{e_k} \to \left(G'_k\right)_{e_{11}}, 1 \leq k \leq 10$.
  - The objects that are in the environment e_{11} should enter in the skin membrane.
    $r_{25} \equiv X'_{i,k}[\,]_0^0 \to [X'_{i,k}]_0^0, 2 \leq i \leq 7, 1 \leq k \leq 10$.
    $r_{26} \equiv W'_{i,k}[\,]_0^0 \to [W'_{i,k}]_0^0, 2 \leq i \leq 7, 1 \leq k \leq 10$.
    $r_{27} \equiv G'_k[\,]_0^0 \to [G'_{i,k}]_0^0, 1 \leq k \leq 10$.
  - The objects that are in the skin membrane must enter the membrane 1.
    $r_{28} \equiv X'_{i,k}[\,]_1^0 \to [X'_{i,k}]_1^0, 2 \leq i \leq 7, 1 \leq k \leq 10$.
    $r_{29} \equiv W'_{i,k}[\,]_1^0 \to [W'_{i,k}]_1^0, 2 \leq i \leq 7, 1 \leq k \leq 10$.
    $r_{30} \equiv G'_k[\,]_1^0 \to [G'_{i,k}]_1^0, 1 \leq k \leq 10$.

– **Module 4** (Feeding and mortality module for the animals that did not have enough food in the previous steps):
We assume that animals only move from the area where they reside to another one if they need to look for resources to survive.

$$r_{31} \equiv [X'_{i,k} G'^{f_i}_j]^0_1 \xrightarrow{(1-k_{i,2})\cdot p_{k,j,i}} [Y'_{i,j}]^-_1, 2 \le i \le 6, 1 \le j,k \le 10.$$
$$r_{32} \equiv [X'_{7,k} X'^{f_7}_{i,j}]^0_1 \xrightarrow{(1-k_{7,2})\cdot p_{k,j,i}} [Y'_{7,j}]^-_1, 2 \le i \le 6, 1 \le j,k \le 10.$$
$$r_{33} \equiv [X'_{7,k} W'^{f_7}_{i,j}]^0_1 \xrightarrow{(1-k_{7,2})\cdot p_{k,j,i}} [Y'_{7,2}]^-_1, 2 \le i \le 6, 1 \le j,k \le 10.$$
$$r_{34} \equiv [X'_{i,k} G'^{f_i}_j]^0_1 \xrightarrow{k_{i,2}\cdot p_{k,j,i}} []^-_1, 2 \le i \le 6, 1 \le j,k \le 10.$$
$$r_{35} \equiv [X'_{7,k} X'^{f_7}_{i,j}]^0_1 \xrightarrow{k_{7,2}\cdot p_{k,j,i}} []^-_1, 2 \le i \le 6, 1 \le j,k \le 10.$$
$$r_{36} \equiv [X'_{7,k} W'^{f_7}_{i,j}]^0_1 \xrightarrow{(1-k_{7,2})\cdot p_{k,j,i}} []^-_1, 2 \le i \le 6, 1 \le j,k \le 10.$$

– **Module 5** (Initial configuration restoration):

- return the objects from e_{11} to the rest of environments.

$$r_{37} \equiv [X'_{i,k}]^-_1 \to []^0_1, 2 \le i \le 6, 1 \le k \le 10.$$
$$r_{38} \equiv [G'_k]^-_1 \to []^0_1, 1 \le k \le 10.$$
$$r_{39} \equiv [W'_{i,k}]^-_1 \to Y'_{i,k}[]^0_1, 2 \le i \le 6, 1 \le k \le 10.$$
$$r_{40} \equiv [Y'_{i,k}]^-_1 \to Y'_{i,k}[]^0_1, 2 \le i \le 6, 1 \le k \le 10.$$
$$r_{41} \equiv [Y'_{i,k}]^0_0 \to Y'_{i,k}[]^0_0, 2 \le i \le 6, 1 \le k \le 10.$$
$$r_{42} \equiv \left(Y'_{i,k}\right)_{e_{11}} \to (Y_i)_{e_k}, 2 \le i \le 7, 1 \le k \le 10.$$
$$r_{43} \equiv Y_i[]^0_0 \to [Y_i]^0_0, 2 \le i \le 7.$$
$$r_{44} \equiv Y_i[]^0_1 \to [Y_i]^+_1, 2 \le i \le 7.$$

- Restore initial configuration.

$$r_{45} \equiv [X_1]^+_1 \to X_1[]^0_1.$$
$$r_{46} \equiv [Y_i]^+_1 \to [X_i]^0_1, 2 \le i \le 7.$$

Following is the list of necessary initial parameters:

– Animal related parameters:

- $k_{i,1}$, $(2 \le i \le 7)$: Fertility ratio for species i (females).
- $k_{i,2}$, $(2 \le i \le 7)$: Survival ratio for species i.
- f_i, $(2 \le i \le 7)$: Amount of food units consumed per year by animal for species i.
- d_i, $(2 \le i \le 7)$: Number of descendants per female and birth for species i.
- $q_{i,j}$, $(2 \le i \le 7,\ 1 \le j \le 10)$: Initial number of individuals of species i in area j.
- $p_{k,j,i}$, $(2 \le i \le 7,\ 1 \le j, k \le 10)$: Probability of movement of species i from the area k to j.

– Grass related parameters:

- h_i, $(1 \le i \le 3)$: Amount of produced grass per hectare in different conditions ($i = 1, 2, 3$ represent unfavorable, average and favorable conditions, respectively).
- m_i, $(1 \le i \le 3)$: Probability of the corresponding condition affecting the production of grass.
- $q_{1,j}$, $(1 \le j \le 10)$: Number of land hectares in area j.

– Besides, $q_{i,11} = 0$, for $1 \le i \le 7$.

4 Binomial block based simulation algorithm

In this section we describe the first simulation algorithm developed for multienvironment probabilistic P systems (Cardona et al. 2011). It follows a strategy based on the binomial distribution and blocks of rules with the same left-hand side.

In general, each simulation step is divided into two main stages: *selection* and *execution*. In the first one, the algorithm decides which rules will be applied, and the number of applications for each one (taking into account their left-hand sides and the available objects in the current configuration). In the second stage, the selected rules are applied, consuming the multisets of the rules' left-hand sides and adding the multisets of the rules' right-hand sides the selected number of times, and possibly changing the polarization of membranes.

Next we describe the selection stage.

Input: A multienvironment probabilistic functional extended P system with active membranes of degree (q, m) with $q \ge 1$, $m \ge 1$, taking T time units, $T \ge 1$.

1: Rules are classified into sets (blocks) so that all the rules belonging to a block have the same left-hand side. Note that rules from different blocks may have overlapping left-hand sides.

2: Let $F_b(N, p)$ be a function that returns a discrete random number within the binomial distribution $B(N, p)$.

3: **for** each step of simulation **do**

4: A random order on the family of blocks is considered.

5: **for all** blocks, according to the considered order **do**

6: A random order on the rules of the block, $\{r_1, \ldots, r_t\}$, is selected.

7: Let us suppose that the common left-hand side is $u[v]^\alpha_i$ and their respective probabilistic constants are $c_{r_1}, \ldots, c_{r_t}$.

8: The highest number N is computed so that u^N appears in the parent membrane of i and v^N appears in membrane i.

9: let $d = 1$

10: **for all** $k(1 \le k \le t-1)$, according to the selected order **do**

11: let c_{r_k} be $\frac{c_{r_k}}{d}$

12: let n_{r_k} be $F(N, c_{r_k})$
13: let N be $N - n_{r_k}$
14: let q be $1 - c_{r_k}$
15: let d be $d*q$
16: **end for**
17: let n_{r_i} be N
18: The pair $\langle r, n_r \rangle$ is added to R_{sel}, which means that each rule r is applied n_r times.
19: **end for**
20: **end for**

Once the list R_{sel} has been calculated, the implementation of the second stage is quite straightforward: first remove all the left-hand sides of the rules, and then add all the right-hand sides (in both cases taking into account the number of times each rule is applied).

This simulation algorithm is useful for most of the cases but it has the next disadvantages:

- It needs to classify the rules by their left-hand side.
- It does not handle rules with intersections on their left-hand sides.
- It does not check the consistency of charges in the selection of rules.
- It does not evaluate probabilistic functions related to rules.

5 Direct non-deterministic distribution algorithm with probabilities (DNDP)

The following is a more efficient simulation algorithm for multienvironment probabilistic P systems inspired by (Nguyen et al. 2009) that overcomes the weak points of the simulation algorithm described above.

Input: A multienvironment functional P system with active membranes of degree (q, m) with $q \geq 1$, $m \geq 1$, taking T time units, $T \geq 1$.

1: C_0 initial configuration of the system
2: **for** $t \leftarrow 0$ to $T - 1$ **do**
3: C'_t C_t
4: *Initialization*
5: *First selection phase*: generates a multiset of *consistent* applicable rules.
6: *Second selection phase*: generates a multiset of *maximal consistent* applicable rules.
7: *Execution of selected rules*.
8: C_{t+1} C'_t
9: **end for**

Initialization

1: R_Π ordered set of rules of Π
2: **for** j 1 to m **do**
3: $R_{E,j}$ ordered set of rules from R_E related to the environment j
4: A_j ordered set of rules from $R_{E,j}$ whose probability at the moment t is > 0
5: LC_j ordered set of pairs $\langle label, charge \rangle$ for all the membranes from C_t contained in the environment j
6: $B_j \leftarrow \emptyset$
7: **for** each $\langle h, \alpha \rangle \in LC_j$ (following the considered order) **do**
8: B_j $B_j \cup$ ordered set of rules $u[v]^\alpha_h \rightarrow u'[v']^\beta_h$ from R_Π whose probability at the moment t is greater than 0 for the environment j
9: **end for**
10: **end for**

First selection phase (*consistency*)

1: **for** j 1 to m **do**
2: R_j *the empty multiset*
3: $D_j \leftarrow A_j \cup B_j$ with a *random order*
4: **for** each $r \in D_j$ (following the considered order) **do**
5: M maximum number of times that r is applicable to C'_t
6: **if** r is *consistent* with the rules in $R^1_j \wedge M > 0$ **then**
7: $N \leftarrow$ maximum number of times that r is applicable to C_t
8: $n \leftarrow \min\{M, F_b(N, p_{r,j}(t))\}$
9: C'_t $C'_t - n \cdot LHS(r)$
10: $R_j \leftarrow R_j \cup \{ < r, n > \}$
11: **end if**
12: **end for**
13: **end for**

Second selection phase (*maximality*)

1: **for** j 1 to m **do**
2: R_j *R_j with an order by the rule probabilities, from highest to lowest*
3: **for** each $< r, n > \in R_j$ (following the selected order) **do**
4: **if** $n > 0 \vee (r$ is *consistent* with the rules in $R^1_j)$ **then**
5: M maximum number of times that r is applicable to C'_t
6: **if** $M > 0$ **then**
7: R_j $R_j \cup \{ < r, M > \}$
8: C'_t $C'_t - M \cdot LHS(r)$
9: **end if**
10: **end if**
11: **end for**
12: **end for**

Execution of selected rules

1: **for** each $< r, n > \in R_j, n > 0$ **do**

2: $C'_t \quad C'_t + n \cdot RHS(r)$

3: Update the electrical charges of C'_t according to $RHS(r)$

4: **end for**

- *Note 1*. If $r : u[v]_i^{\alpha} \rightarrow u'[v']_i^{\alpha'}$ is a rule of the system, then $u[v]_i^{\alpha}$ is the left-hand side of the rule r (denoted by $LHS(r)$), and $u'[v']_i^{\alpha'}$ is its right-hand side (denoted by $RHS(r)$).
- *Note 2*. Two rules r and r' with the same labels and electrical charges for $LHS(r)$ and $LHS(r')$ are *consistent* if $RHS(r)$ and $RHS(r')$ have the *same* electrical charges.
- *Note 3*. $p_{r,j}(t)$ indicates the probability associated to rule r for the P system located in the environment j, at the moment t.

6 Simulation results

The inherent randomness in the dynamics of ecosystems makes it unfeasible to address the formal validation of models that attempt to reproduce their behavior. It is therefore necessary to carry out an experimental validation, by comparing results generated by simulation tools against experimental data obtained directly from the real ecosystem. Moreover, once a model is (experimentally) validated, it is possible to use the software tool to analyze the dynamics of the real-life process for different virtual scenarios that could be interesting for the experts in order to formulate plausible hypotheses.

We have used the P-Lingua (García-Quismondo et al. 2010; http://www.p-lingua.org/) programming language and the *pLinguaCore* (García-Quismondo et al. 2010; http://www.p-lingua.org/wiki/index.php/PLinguaCore) library in order to simulate the model. P-Lingua is a programming language to define P systems in an easy-to-learn, parametric and modular way, and pLinguaCore is a Java library under GNU GPL license (http://www.gnu.org/copyleft/gpl.html) which implements several simulation algorithms for P systems. In particular, the current release of pLinguaCore includes implementations of the two algorithms described in the previous sections.

A new graphic user interface (GUI) have been developed over pLinguaCore (Pérez-Hurtado et al. 2010). It allows the edition of the initial parameters of the ecosystem and the collection of simulation results. Figures 2, 3, 4, 5, and 6 show how the parameters are introduced by using the GUI. This GUI and all the files related to the simulator can be downloaded from http://www.p-lingua.org/.

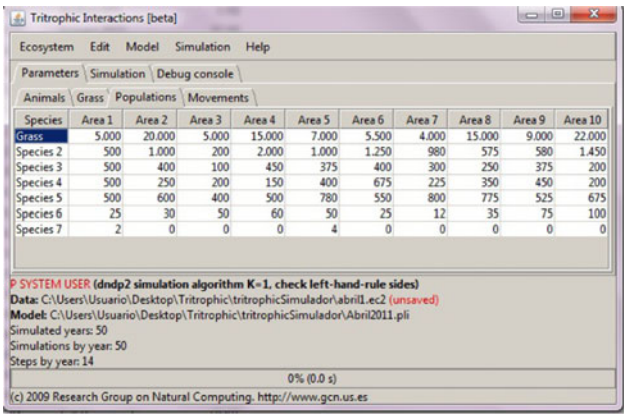


Fig. 2 Number of animals of each species and grass surface

In order to compare the two previously described algorithms, we have considered a theoretical ecosystem with three trophic levels composed by 6 species of animals (5 ungulates and 1 carnivore that predates on them). The ecosystem is geographically divided into 10 areas. Figure 2 shows the initial number of animals of each species and the surface of each of the 10 areas (Grass).

The following assumptions have been accepted:

- The base diet of ungulates is the grass.
- The carnivorous species (referred to as "Species 7" in the table) feeds on ungulates (Species 2–6 in the table correspond to ungulates).
- Grass production depends on environmental conditions.
- When ungulates do not have enough food in the area where they reside at the moment, they may move to another adjacent area.

The ungulates are herbivores, and usually when reproducing, each fertile female has one descendent. However, carnivores in general (as might be the Wolf) have more than one descendent, usually 2–3 or more. Ungulates are supposed to belong to different species having different weights, and therefore the amount of biomass provided by an ungulate for the carnivores to eat should depend on the species. In this hypothetical example and for the sake of simplicity we have assumed that one carnivore needs ten ungulates each year for survival. The values taken for the biological parameters are shown in Fig. 3.

The base diet of ungulates is the grass produced within the ecosystem. Depending on climatic conditions production can vary, and in this example three different situations are considered. The first situation corresponds to a year with low grass production, the second represents the common (average) case, and the third would be a year with production values higher than usual. Each of these three situations may occur randomly each year, according to associated probabilities. Figure 4 shows the values taken in this case.

When there is shortage of resources on a given environment, the model gives the animals a chance to migrate to another environment. The carnivorous species are always supposed to move, and they can hunt ungulates from any other area. On the contrary, ungulates can only move to adjacent areas (eventually they may stay where they are). The adjacency relation of the 10 geographic zones considered is graphically depicted in Fig. 5.

Rules that take care of this movements for finding food are listed in Module 4. Figure 6 shows the screen of the simulator with some of the values of the related parameters.

We have simulated the evolution of this ecosystem for 50 years, and for each year 50 repetitions were made. The simulator provides not only graphical output, but also numerical results. The results from the point of view of ecology have been the same irrespectively of the implemented algorithm. Figure 7 shows results for the area 1,

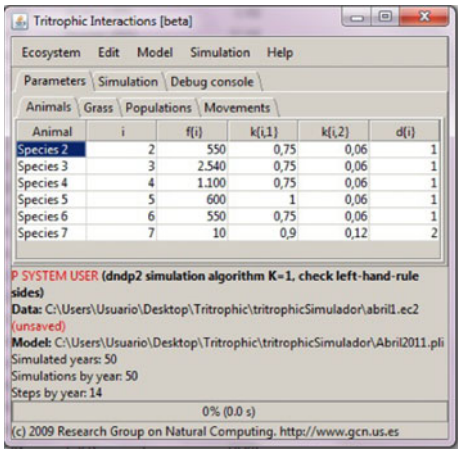


Fig. 3 Biological parameters

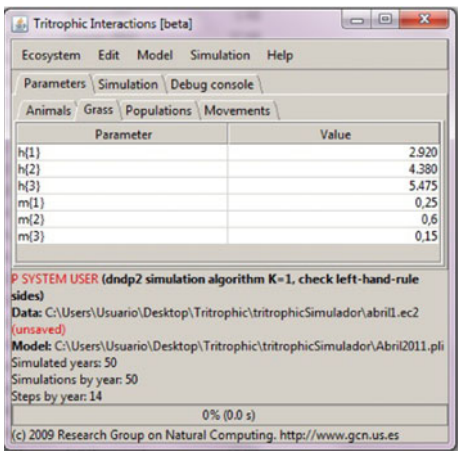


Fig. 4 Parameters related to grass

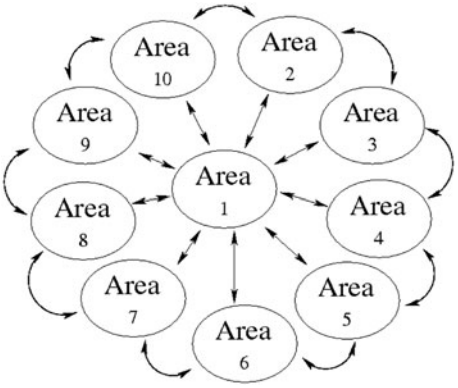


Fig. 5 Adjacency graph of the ten geographic zones considered

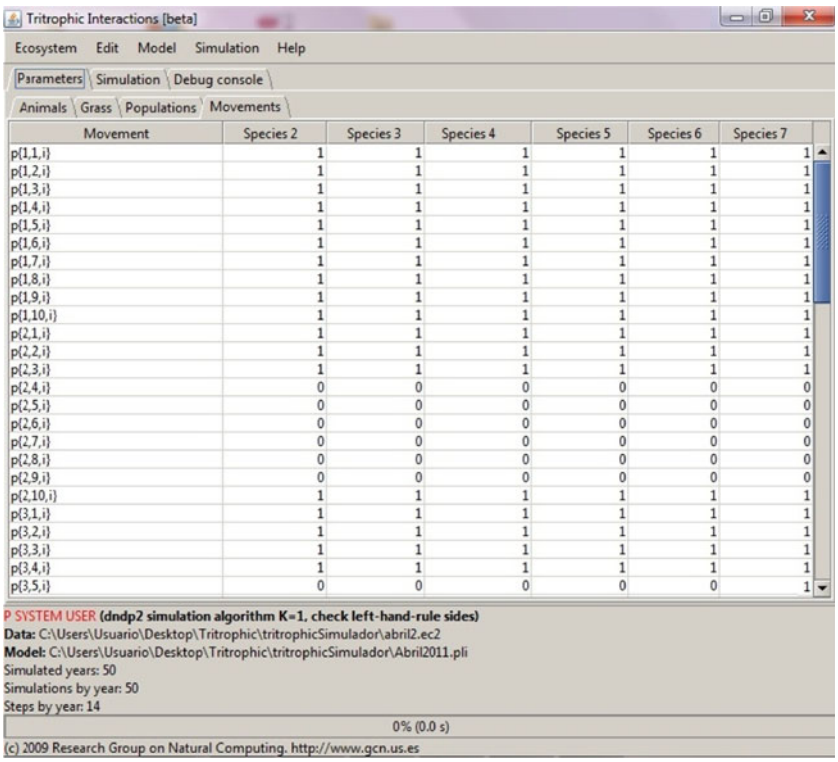
where the initial number of individuals was the same for four of the five ungulates. Moreover, for three of these ungulates the same biological parameters were introduced, while in the other case (species 5) the probability of reproduction was higher. This difference is reflected in the number of animals to which each species is stabilized.

In Fig. 8 the total number of ungulates and carnivores in the ecosystem are shown. At first, populations of ungulates and carnivores grow, but one can notice that there is a small offset between the years when the maximum of ungulates and carnivores are reached. At some point the necessary amount of ungulates required to feed the carnivores is larger than the actual number of individuals. Then the decline of ungulates starts, and shortly after that of the carnivores, which fails to overcome. Note that carnivores can migrate between different areas in an arbitrary manner when they do not have enough resources, but nevertheless they can go to settings where there is no food and die.

Figure 9 shows that between 19–23 the number of carnivores exceeds the maximum number of carnivores that could survive according to the available biomass on the ecosystem, and this fact causes their extinction. Obviously, if the number of carnivores decreases to 0, the ungulates will increase their population and thus there will be more biomass on the ecosystem. Consequently the line representing the maximum number of carnivores increases as well.

To study whether the new algorithm, DNDP, improves the previous one, both algorithms were run on two different scenarios, measuring the time required to simulate 50 years with 50 repetitions per year. The first scenario consists of 10 environments, but initially there are only animals in area 1. The second scenario used is discussed in the previous results. Table 1 shows the values obtained on seven simulations. On average, DNDP shows a reduction of around 5-6% on the execution time with respect to the Binomial algorithm.

Fig. 6 $p(x, y, i) = 1$ means that the areas x and y are geographically connected



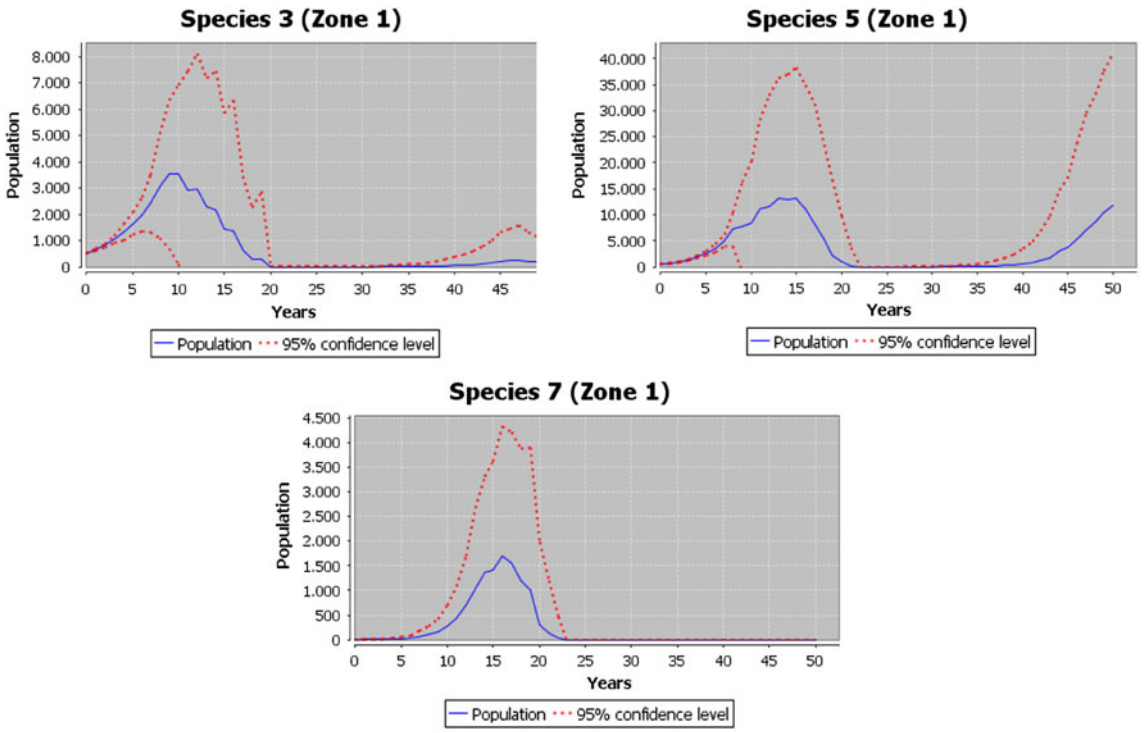


Fig. 7 Evolution of species

7 Conclusions

The results presented in the previous section show an improvement ranging from 25.5% to 27.5% on the performance of DNDP algorithm with respect to the Binomial algorithm (as far as the simulation time is concerned). However, the presented tritrophic ecosystem was not designed specifically to be used in comparative studies.

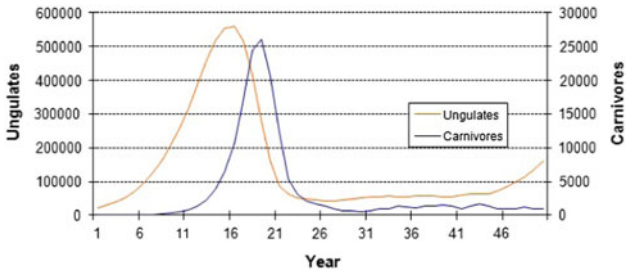


Fig. 8 Total number of ungulates and carnivores per year

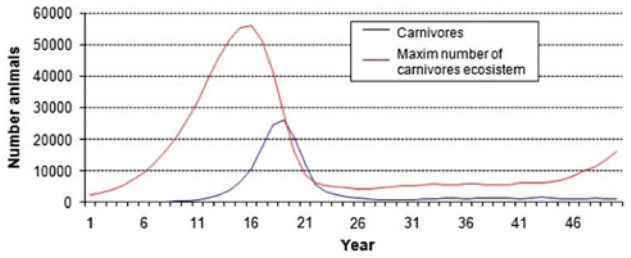


Fig. 9 Total number of carnivores and maximum possible number of carnivores

Table 1 Time (s) of a 50 years simulation, running 50 repetitions per year

Algorithm	Scenario 1		Scenario 2	
	DNDP	Binomial	DNDP	Binomial
Simulation 1	279,625	361,828	252,427	353,723
Simulation 2	277,213	359,083	259,228	347,592
Simulation 3	261,425	371,693	256,284	360,475
Simulation 4	266,997	360,677	259,332	366,052
Simulation 5	270,313	358,339	252,974	344,262
Simulation 6	266,993	357,915	257,198	343,839
Simulation 7	261,051	358,159	248,568	348,813
Average	269,088	361,099	255,144	352,108
Deviation	6,661	4,526	3,685	7,822

In order to draw significant conclusions on the speed-up achieved with DNDP one needs to simulate a more demanding case study.

For instance, in Martínez et al. (2010) a parallel implementation of the DNDP algorithm is presented, making use of multi-thread programming in Java. In that paper, the algorithm is tested on a P system having 16 environments and 4000 rules, and in this case the obtained improvement grows up to 84% (again, w.r.t. simulation time).

To summarize, this paper presents a virtual ecosystem that tries to balance simplicity and biological relevance. We believe that it was necessary to define a scenario smaller than real case studies (on the size of the alphabet, number of rules, number of parameters, etc) but complex enough to become an illustrative example for the ingredients of the modeling framework (multienvironment functional probabilistic P system with active membranes) and for the possible simulation algorithms that could be implemented.

On one hand, despite its simplicity, it has been tailored keeping as close as possible to ecological reality. Thus, it can be very useful as a first case study to introduce researchers from other areas to the modeling framework in membrane computing.

On the other hand, this hypothetical ecosystem can be used as a benchmark case study for algorithms (and programs) of this kind, as it was illustrated in the paper. As future work, a particularly interesting experiment is to study the performance of simulation algorithms running on parallel architectures (e.g. GPUs) when they receive large case studies as input.

Acknowledgments The authors acknowledge the support of the project TIN2009-13192 of the Ministerio de Educación y Ciencia of Spain, cofinanced by FEDER funds, and the support of the "Proyecto de Excelencia con Investigador de Reconocida Valía" of the Junta de Andalucía under grant P08-TIC04200.

References

Besozzi D, Cazzaniga P, Pescini D, Mauri G (2008) Modelling metapopulations with stochastic membrane systems. BioSystems 91(3):499–514

Cardona M, Colomer MA, Pérez-Hurtado I, Pérez-Jiménez MJ, Sanuy D, Margalida A (2010) A P system based model of an ecosystem of some scavenger birds. In: Membrane computing. 10th international workshop, WMC 2009. Lecture notes in computer science, vol 5957, pp 182–195

Cardona M, Colomer MA, Margalida A, Palau A, Pérez-Hurtado I, Pérez-Jiménez MJ, Sanuy D (2011) A computational modeling for real ecosystems based on P systems. Nat Comput. doi: 10.1007/s11047-010-9191-3

García-Quismondo M, Gutiérrez-Escudero R, Pérez-Hurtado I, Pérez-Jiménez MJ, Riscos-Núñez A (2010) An overview of P-Lingua 2.0. In: Membrane computing. 10th international workshop, WMC 2009. Lecture notes in computer science, vol 5957, pp 264–288

GPL license. http://www.gnu.org/copyleft/gpl.html

Martínez MA, Pérez-Hurtado I, Pérez-Jiménez MJ, Riscos-Núñez A, Colomer MA (2010) A new simulation algorithm for multienvironment probabilistic P systems. In: Li K, Tang Z, Li R, Nagar AK, Thamburaj R (eds) Proceedings 2010 IEEE fifth international conference on Bio-inpired Computing: Theories and Applications (BIC-TA 2010), vol 1. IEEE Press, Changsha, China, 23-26 September 2010, pp 59–68

Nguyen V, Kearney D, Gioiosa G (2009) An algorithm for nondeterministic object distribution in P systems and its implementation in hardware. In: Membrane computing. 9th international workshop, WMC 2008. Lecture notes in computer science, vol 5391, pp 325–354

Păun Gh, Rozenberg G, Salomaa A (eds) (2010) The Oxford handbook of membrane computing. Oxford University Press, Oxford

Pérez-Jiménez MJ, Romero FJ (2006) P systems, a new computational modelling tool for systems biology. In: Transactions on computational systems biology VI. Lecture notes in bioinformatics, vol 4220, pp 176–197

Romero FJ, Pérez-Jiménez MJ (2008) Modelling gene expression control using P systems: the Lac Operon, a case study. BioSystems 91(3):438–457

Pérez-Hurtado I, Valencia L, Pérez-Jiménez MJ, Colomer MA, Riscos-Núñez A (2010) MeCoSim: a general purpose software tool for simulating biological phenomena by means of P Systems. In: Li K, Tang Z, Li R, Nagar AK, Thamburaj R (eds) Proceedings 2010 IEEE fifth international conference on Bio-inpired Computing: Theories and Applications (BIC-TA 2010), vol 1. IEEE Press, Changsha, China, 23–26 September 2010, pp 637–643

The P-Lingua website. http://www.p-lingua.org/

The pLinguaCore library website. http://www.p-lingua.org/wiki/index.php/PLinguaCore