Generalized Communicating P Systems
Working in Fair Sequential Mode

Antoine Spicher
LACL, Département Informatique
Université Paris Est
61 av. gén. de Gaulle, 94010, Créteil, France
antoine.spicher@u-pec.fr

Sergey Verlan
LACL, Département Informatique
Université Paris Est
61 av. gén. de Gaulle, 94010, Créteil, France
verlan@u-pec.fr

In this article we consider a new derivation mode for generalized communicating P systems (GCPS) corresponding to the functioning of population protocols (PP) and based on the sequential derivation mode and a fairness condition. We show that PP can be seen as a particular variant of GCPS. We also consider a particular stochastic evolution satisfying the fairness condition and obtain that it corresponds to the run of a Gillespie’s SSA. This permits to further describe the dynamics of GCPS by a system of ODEs when the population size goes to the infinity.

1 Introduction

The notion of a generalized communicating P system was introduced in [22], with the aim of providing a common generalization of various purely communicating models in the framework of P systems.

A generalized communicating P system, or a GCPS for short, corresponds to a hypergraph where each node is represented by a cell and each edge is represented by a rule. Every cell contains a multiset of objects which – by communication rules – may move between the cells. The form of a communication rule is \((a,i)(b,j) \rightarrow (a,k)(b,l)\) where \(a\) and \(b\) are objects and \(i, j, k, l\) are labels identifying the input and the output cells. Such a rule means that an object \(a\) from cell \(i\) and an object \(b\) from cell \(j\) move synchronously to cell \(k\) and cell \(l\), respectively. In this respect, the model resembles the Petri Net formalism [18] where tokens from various input places come along together to fire a given transition and then fork out to destination places, see [22, 4] for more details.

Depending on the communication rules form, several restrictions on communication rules (modulo symmetry) can be introduced. Due to the simplicity of their rules, the generative power of such restricted systems is of particular interest and it has been studied in detail. In [22, 8, 7, 17] it was proved that eight of the possible nine restricted variants (with respect to the form of rules) are able to generate any recursively enumerable set of numbers; in the ninth case only finite sets of singletons can be obtained. Furthermore, these systems even with relatively small numbers of cells and simple underlying (hypergraph) architectures are able to achieve this generative power. In [7] a further restriction is introduced by considering that the alphabet of objects is a singleton (like in Petri Nets) and it is shown that the computational completeness can be achieved in four of the restricted variants.

Population protocols (PP) have been introduced in [1] (see [3] for a survey) as a model of sensor networks consisting of very limited mobile agents with no control over their own movement. A population protocol corresponds to a collection of anonymous agents, modelled by finite automata, that interact with one another to carry out computations, by updating their states, using some rules. Their computational power has been investigated under several hypotheses in most of the cases restricted to finite size populations. In particular, predicates stably computable in the original model have been characterized as those
definable in Presburger arithmetic. The article [5] studies the convergence of PP when the population size goes to the infinity.

The evolution of a PP follows a particular fairness condition: an execution is fair if for all configurations $C$ that appear infinitely often in the execution, if $C$ is predecessor of a configuration $C'$, then $C'$ appears infinitely often in the execution. We consider such a condition in the case of GCPS systems and obtain a new derivation mode which we call fair sequential mode (fs-mode). We further study the dynamic behaviour of the system in this mode. Among several possible evolution strategies we consider a stochastic strategy that satisfies the fairness condition and we obtain that the evolution of the system corresponds to a run of the Gillespie stochastic simulation algorithm (SSA). Using the correspondence between SSA and ODEs (assuming mass-action kinetics) we show that the dynamics of the system corresponds to a run of the Gillespie stochastic simulation algorithm (SSA) for the system in this mode. Among several possible evolution strategies we consider the converse problem and we give sufficient conditions for a system of ODEs to be represented by a GCPS system working in concentration-depended stochastic implementation of the fs-mode. We consider several examples of GCPS simulating Lotka-Volterra (predator-prey) behaviour or computing approximations of algebraic numbers.

2 Background

In this section we recall some basic notions and notations used in membrane computing, formal language theory and computability theory. For further details and information the reader is referred to [16, 17, 19].

An alphabet is a finite non-empty set of symbols. For an alphabet $V$, we denote by $V^*$ the set of all strings over $V$, including the empty string, $\lambda$. The length of the string $x \in V^*$ is the number of symbols which appear in $x$ and it is denoted by $|x|$. The number of occurrences of a symbol $a \in V$ in $x \in V^*$ is denoted by $|x|_a$. If $x \in V^*$ and $U \subseteq V$, then we denote by $|x|_U$ the number of occurrences of symbols from $U$ in $x$.

A finite multiset over $V$ is a mapping $M : V \rightarrow \mathbb{N}$; $M(a)$ is said to be the multiplicity of $a$ in $M$ ($\mathbb{N}$ denotes the set of non-negative integers.) A finite multiset $M$ over an alphabet $V$ can be represented by all permutations of a string $x = a_1^{M(a_1)}a_2^{M(a_2)}\ldots a_n^{M(a_n)} \in V^*$, where $a_j \in V$, $1 \leq j \leq n$; $x$ represents $M$ in $V^*$. If no confusion arises, we also may use the customary set notation for denoting multisets. The size of a finite multiset $M$, represented by $x \in V^*$ is defined as $\Sigma_{a \in V}|x|_a$.

2.1 P Systems

Next we recall the basic definitions concerning generalized communicating P systems [22].

**Definition 1.** A generalized communicating P system (a GCPS) of degree $n$, where $n \geq 1$, is an $(n + 4)$-tuple $\Pi = (O, E, w_1, \ldots, w_n, R, h)$ where

1. $O$ is an alphabet, called the set of objects of $\Pi$;
2. $E \subseteq O$; called the set of environmental objects of $\Pi$;
3. $w_i \in O^*$, $1 \leq i \leq n$, is the multiset of objects initially associated with cell $i$;
4. $R$ is a finite set of interaction rules (or communication rules) of the form $(a, i)(b, j) \rightarrow (a, k)(b, l)$, where $a, b \in O$, $0 \leq i, j, k, l \leq n$, and if $i = 0$ and $j = 0$, then $\{a, b\} \cap (O \setminus E) \neq \emptyset$; i.e., $a \notin E$ and/or $b \notin E$;
5. $h \in \{1, \ldots, n\}$ is the output cell.
The system consists of \( n \) cells, labelled by natural numbers from 1 to \( n \), which contain multisets of objects over \( O \); initially cell \( i \) contains multiset \( w_i \) (the initial contents of cell \( i \) is \( w_i \)). We distinguish an additional special cell, labelled by 0, called the environment. The environment contains objects of \( E \) in an infinite number of copies.

The cells interact by means of the rules \((a,i)(b,j) \to (a,k)(b,l)\), with \( a,b \in O \) and \( 0 \leq i,j,k,l \leq n \). As the result of the application of the rule, object \( a \) moves from cell \( i \) to cell \( k \) and \( b \) moves from cell \( j \) to cell \( l \). If two objects from the environment move to some other cell or cells, then at least one of them must not appear in the environment in an infinite number of copies. Otherwise, an infinite number of objects can be imported in the system in one step.

A configuration of a GCPS \( \Pi \), as above, is an \((n+1)\)-tuple \((z_0,z_1,\ldots,z_n)\) with \( z_0 \in (O \setminus E)^* \) and \( z_i \in O^* \), for all \( 1 \leq i \leq n \); \( z_0 \) is the multiset of objects present in the environment in a finite number of copies, whereas, for all \( 1 \leq i \leq n \), \( z_i \) is the multiset of objects present inside cell \( i \). The initial configuration of \( \Pi \) is the tuple \((\lambda,w_1,\ldots,w_n)\).

Given a multiset of rules \( \mathcal{R} \) over \( R \) and a configuration \( u = (z_0,z_1,\ldots,z_n) \) of \( \Pi \), we say that \( \mathcal{R} \) is applicable to \( u \) if all its elements can be applied simultaneously to the objects of multisets \( z_0,z_1,\ldots,z_n \) such that every object is used by at most one rule. Then, for a configuration \( u = (z_0,z_1,\ldots,z_n) \) of \( \Pi \), a new configuration \( u' = (z'_0,z'_1,\ldots,z'_n) \) is obtained by applying the rules of \( R \) in a non-deterministic maximally parallel manner: taking an applicable multiset of rules \( \mathcal{R} \) over \( R \) such that the application of \( \mathcal{R} \) results in configuration \( u' = (z'_0,z'_1,\ldots,z'_n) \) and there is no other applicable multiset of rules \( \mathcal{R}' \) over \( R \) which properly contains \( \mathcal{R} \).

It is also possible to replace the maximally parallel strategy of rule application by other strategies, called derivation modes (in the context of the present paper, the terms mode and strategy are used indifferently). A derivation mode lies in the heart of the semantics of P systems and it permits to specify which multiset among different possible applicable multisets of rules can be applied. When P systems were introduced, only the maximally parallel derivation mode was considered which states that corresponding multisets should be maximal, i.e., non-extensible. With the appearance of the minimal parallel derivation mode \([6]\) the concept of the derivation mode had to be precisely defined and \([10]\) presents a framework that permits to easily define different derivation modes.

One application of a multiset of rules satisfying the conditions of a derivation mode represents a transition in \( \Pi \) from configuration \( u \) to configuration \( u' \). A transition sequence is said to be a successful generation by \( \Pi \) if it starts with the initial configuration of \( \Pi \) and ends with a halting configuration, i.e., with a configuration where no further transition step can be performed.

We say that \( \Pi \) generates a non-negative integer \( n \) if there is a successful generation by \( \Pi \) such that \( n \) is the size of the multiset of objects present inside the output cell in the halting configuration. The set of non-negative integers generated by a GCPS \( \Pi \) in this way is denoted by \( N(\Pi) \). It is also possible to use GCPS as acceptors, in this case an input multiset is accepted if the system halts on it.

In \([22]\) it is shown that GCPS are able to generate all recursively enumerable languages. Moreover this result can be obtained by using various restrictions on the type of rules (i.e. induced hypergraph structures), on the number of membranes and on the cardinality of the alphabet. We refer to \([22,8,7]\) for more details.

If the cardinality of the alphabet \( O \) is equal to one, then we refer to the corresponding symbol as a token (denoted by \( \bullet \)). Hence, we assume that \( O = \{\bullet\} \). We observe that such systems are similar to Petri Nets having a restricted topology. This is especially visible if a graphical notation is used. However, the maximal parallelism and the concept of the environment are specific to P systems, so we place this study in the latter framework. A converse study of P systems from the point of view of Petri Nets can be found in \([11]\). For more details on Petri Nets and membrane computing we also refer to \([17]\).
In this article we shall consider the dynamics of the configuration of GCPS, so we are no more interested in computation (and halting evolutions).

### 2.2 Population Protocols

We give below the definition as it appears in [5]. A protocol is given by $(Q, \Sigma, 1, \omega, \delta)$ with the following components. $Q$ is a finite set of states. $\Sigma$ is a finite set of input symbols. $1: \Sigma \rightarrow Q$ is the initial state mapping, and $\omega: Q \rightarrow \{0, 1\}$ is the individual output function. $\delta \subseteq Q^4$ is a joint transition relation that describes how pairs of agents can interact. Relation $\delta$ is sometimes described by listing all possible interactions using the notation $(q_1, q_2) \rightarrow (q_1', q_2')$, or even the notation $q_1 q_2 \rightarrow q_1' q_2'$, for $(q_1, q_2, q_1', q_2') \in \delta$ (with the convention that $(q_1, q_2) \rightarrow (q_1, q_2)$ when no rule is specified with $(q_1, q_2)$ in the left hand side.)

Computations of a protocol proceed in the following way. The computation takes place among $n$ agents, where $n \geq 2$. A configuration of the system can be described by a vector of all the agent’s states. The state of each agent is an element of $Q$. Because agents with the same states are indistinguishable, each configuration can be summarized as an unordered multiset of states, and hence of elements of $Q$.

Each agent is given initially some input value from $\Sigma$: each agent’s initial state is determined by applying $1$ to its input value. This determines the initial configuration of the population.

An execution of a protocol proceeds from the initial configuration by interactions between pairs of agents. Suppose that two agents in state $q_1$ and $q_2$ meet and have an interaction. They can change into state $q_1'$ and $q_2'$ if $(q_1, q_2, q_1', q_2')$ is in the transition relation $\delta$. If $C$ and $C'$ are two configurations, we write $C \rightarrow C'$ if $C'$ can be obtained from $C$ by a single interaction of two agents: this means that $C$ contains two states $q_1$ and $q_2$ and $C'$ is obtained by replacing $q_1$ and $q_2$ by $q_1'$ and $q_2'$ in $C$, where $(q_1, q_2, q_1', q_2') \in \delta$. An execution of the protocol is a (potentially infinite) sequence of configurations $C_0, C_1, C_2, \ldots$, where $C_0$ is an initial configuration and $C_i \rightarrow C_{i+1}$ for all $i \geq 0$. An execution is fair if for all configurations $C$ that appears infinitely often in the execution, if $C \rightarrow C'$ for some configuration $C'$, then $C'$ appears infinitely often in the execution.

At any point during an execution, each agent’s state determines its output at that time. If the agent is in state $q$, its output value is $\omega(q)$. The configuration output is 0 (respectively 1) if all the individual outputs are 0 (respectively 1). If the individual outputs are mixed 0s and 1s then the output of the configuration is undefined.

Let $p$ be a predicate over multisets of elements of $\Sigma$. Predicate $p$ can be considered as a function whose range is $\{0, 1\}$ and whose domain is the collection of these multisets. The predicate is said to be computable by the protocol if, for every multiset $I$, and every fair execution that starts from the initial configuration corresponding to $I$, the output value of every agent eventually stabilizes to $p(I)$.

The following was proved in [1, 2]:

**Theorem 1** ([1, 2]). *A predicate is computable in the population protocol model if and only if it is semilinear.*

Recall that semilinear sets are known to correspond to predicates on counts of input agents definable in first-order Presburger arithmetic [15].

### 2.3 Gillespie Algorithm

A usual abstraction in the simulation of biochemical systems consists in considering the system (e.g., a bacterium) as a homogeneous chemical solution where the reactions of the model are taking place. D.T. Gillespie has proposed in [12] an algorithm for producing the trajectories of such a chemical system.
by computing the next reaction and the elapsed time since last reaction occurred. Let $\mu$ be a chemical reaction. The probability that $\mu$ takes place during an infinitesimal time step is proportional to:

- $c_\mu$, the stochastic reaction constant of reaction $\mu$;
- $h^S_\mu$, the number of distinct molecular combinations that can activate reaction $\mu$; it depends on the current chemical state $S$;
- $d\tau$, the length of the time interval.

Gillespie proved that the probability $P(\tau, \mu | S) d\tau$ that, being in a chemical state $S$, the next reaction will be of type $\mu$ and will occur in the time interval $(t + \tau, t + \tau + d\tau)$ is:

$$P(\tau, \mu | S) d\tau = a^S_\mu e^{-a^S_0} d\tau$$

where $a^S_\mu = c_\mu h^S_\mu$ is called the propensity of reaction $\mu$, and $a^S_0 = \sum \nu a^S_\nu$ is the combined propensity of all reactions.

This probability leads to the first straightforward Gillespie’s exact stochastic simulation algorithm (SSA) called the first reaction method. From a chemical state $S$, it consists in choosing an elapsed time $\tau$ for each reaction $\mu$ according to the probability $P(\tau, \mu | S)$. The reaction with the lowest elapsed time is selected and applied on the system making its state evolve. A new probability distribution is then computed for this new state and the process is iterated.

The Gillespie’s SSA gives a way to simulate a continuous-time Markov chain with the states corresponding to the states of the system and with transitions between states corresponding to a single occurrence of a reaction. The probability for a transition between two states $S$ and $S'$ corresponding to the application of rule $\mu$ is defined as $a^S_\mu / a^S_0$. In the following, we drop the mention of the current state $S$ in these notations.

3 Fair Sequential Derivation Mode

In this section we are interested in the relation between PP and GCPS. We show that in terms of structure PP and GCPS are quite similar, the main differences concern the environment and the derivation mode. We define a new fair sequential mode for GCPS and hence we are able to encode any PP in a GCPS w.r.t. their dynamics. We then remark that GCPS with stochastic and Gillespie-like strategies are part of this new class of GCPS and we propose their use for simulations of population behaviours.

It can be easily seen that both PP and GCPS are particular instances of multiset rewriting. Indeed, in both cases the underlying data structure is multiset (obtained in a direct way for PP and by attaching the indices of membranes to the objects in GCPS) and the evolution rules clearly correspond to multiset rewriting rules with both left hand and right hand sides of size two. So, the translation of a PP to a one-symbol GCPS can be easily done as follows. Given a PP with set of states $Q$ (for convenience we suppose that $Q = \{1, \ldots, n\}$) and transition relation $\delta$ in an initial configuration $C_0$, the corresponding GCPS $\Pi = (O, E, w_1, \ldots, w_n, R, 1)$ can be defined as:

- $O = E = \{\bullet\}$,
- $w_q = \bullet^k$, $k = |C_0|_q$ for any $q \in Q$,
- $R = \{(\bullet, q_1)(\bullet, q_2) \rightarrow (\bullet, q_1')(\bullet, q_2') | q_1q_2 \rightarrow q_1'q_2' \in \delta\}$.

1Evaluating the stochastic constants is one of the key issues in stochastic simulations of biochemical reactions.
The above system encodes each state \( q \) of PP by a token • present in membrane labelled by \( q \). Since we are interested in the dynamics of the system, no output membrane is necessary. The above construction covers the core of PP and to obtain the complete equivalence encoding and decoding functions \( \iota \) and \( \omega \) shall be used in the same way.

It is not possible to do a similar encoding of GCPS with PP because PP always deal with finite multisets and GCPS can use the infinite multiset corresponding to the environment. However, any GCPS having no rule involving the environment can be translated to PP in a similar way.

We remark that the biggest difference between PP and GCPS is given by the evolution step, i.e., by the derivation mode. For GCPS, mainly the maximally parallel derivation mode is investigated with several attempts to investigate asynchronous or minimally parallel derivation mode, see [17] for more details. The derivation mode of PP is very particular – it corresponds to a sequential strategy where only one rule is applied at each step, like in Petri Nets, but with an additional fairness condition.

We can consider such a strategy in GCPS case as well. More precisely we consider fair computations: a GCPS computation is fair, if for any configuration \( u \) that appears infinitely often in the computation, then any configuration \( u' \), such that \( u \Rightarrow u' \) in sequential application, also appears infinitely often. We shall call such computational strategy a fair sequential derivation mode (shortly fs-mode).

From these considerations, it is trivial to observe that PP are similar to GCPS in fs-mode with only one symbol in the alphabet. If we consider an encoding function \( \iota \) like for PP and the halting condition corresponding to the stabilization of the \( \omega \)-image of the configuration, then as an immediate consequence of [1, 2], we obtain that any GCPS working in fs-mode and that does not have any rule involving the environment can only accept semilinear sets.

Conversely, we also obtain that any PP working in maximally parallel mode (i.e., a maximally parallel number of interactions can happen at each step) are computationally complete if the number agents in some particular state \( q_0 \) is going to the infinity.

From now on, we only speak of PP in terms of their associated one-symbol GCPS in fs-mode.

### 3.1 FS-Mode and Stochastic Evolution

Although powerful the definition of the fairness remains obscure. Let try to clarify it. When assuming that the number of configurations is finite (this is the case for classical PP for example), the definition can be easily rephrased as follows: a computation \( u_0 \Rightarrow u_1 \Rightarrow \ldots \) is fair if

- there exists a non-negative integer \( N \) such that configuration \( u_N \) belongs to a terminal strongly connected component of the state graph\(^2\) and
- any state of this terminal strongly connected component appears infinitely often in the execution.

There are many possible evolutions of the system in the fs-mode. One example of such an evolution is to choose at each step a rule that leads to a configuration that either never was visited previously or was not visited for some time greater than \( k \), \( k > 0 \) (if possible).

Among all possible evolutions, Markovian processes feature prominently since they respect the fairness condition (it is well-known that Markovian processes leave non-terminal strongly connected components with probability 1), they do not require any history or global knowledge on the state space, and they provide a modelling tool useful in many domains (like in the simulation of population behaviours or in distributed algorithmics). Such a Markovian process corresponds to a labeling of each state graph arrow \( u \Rightarrow^p u' \) by a static probability \( p \) that only depends on configuration \( u \). Here are two examples of such Markovian processes:

\[^2\text{In this directed graph, nodes correspond to the configurations } u, \text{ and two nodes } u \text{ and } u' \text{ are directly linked if } u \Rightarrow u'.\]
1. **Equiprobable evolutions:** $u \Rightarrow \frac{1}{k_u} u'$ where $k_u$ denotes the cardinality of the set $\{u' \mid u \Rightarrow u'\}$.

2. **Concentration-dependent evolutions:** $u \Rightarrow p_r u'$ where $r$ denotes the applied rule and $p_r$ is proportional to $h_r$, the number of distinct combinations of tokens that activate $r$, with a proportionality coefficient that only depends on $r$. Assuming that $r = (\bullet, q_1)(\bullet, q_2) \rightarrow (\bullet, q'_1)(\bullet, q'_2)$, the number $h_r$ is given by

$$h_r = \begin{cases} |u| q_1 |u| q_2 & \text{if } q_1 \neq 0, q_2 \neq 0, q_1 \neq q_2 \\ |u| q_1 (|u| q_1 - 1) & \text{if } q_1 \neq 0, q_1 = q_2 \\ |u| q_1 & \text{if } q_2 = 0 \\ |u| q_2 & \text{if } q_1 = 0 \\ \end{cases}$$  \hfill (1)

The two last cases hold when the environment (containing an infinite number of tokens) is involved in the rule.

### 3.2 FS-Mode GCPS modelling Population Dynamics

Assuming that for a given rule proportionality coefficients are the same for all configurations, the concentration-dependent strategy directly corresponds to a run of the Gillespie’s SSA. Thus, we advocate that GCPS in fs-mode provide a good theoretical tool for studying population behaviours.

A paradigmatic example illustrating how GCPS allows a well suited specification of population behaviours consists of the description of a process inspired by the Lotka-Volterra model.

#### The Lotka-Volterra Model

The Lotka-Volterra process was introduced by Lotka as a model of coupled auto-catalytic chemical reactions, and was investigated by Volterra as a model for studying an ecosystem of predators and preys \[^9\]. This model specifies how two coupled populations (of chemicals or individuals) $Y_1$ (the preys) and $Y_2$ (the predators) behave. In \[^12\], D.T. Gillespie proposes the study of this system derived from the following ODEs

$$\begin{align*}
\frac{dY_1}{dt} &= (c_1 - c_2 Y_2)Y_1 \\
\frac{dY_2}{dt} &= (c_2 Y_1 - c_3)Y_2
\end{align*}$$  \hfill (2)

Equivalently, the following chemical reactions specify a model whose behaviour is described by ODEs system \[^2\]. The dynamics of these reactions is conveniently characterized using the predator-prey interpretation.

$Y_1 \xrightarrow{c_1} 2Y_1 \\
Y_1 + Y_2 \xrightarrow{c_2} 2Y_2 \\
Y_2 \xrightarrow{c_3} . \hfill (3)$

The first rule states that a prey $Y_1$ reproduces. The second rule states that a predator $Y_2$ reproduces after feeding on prey $Y_1$. Finally, the last rule specifies that predators $Y_2$ die of natural causes. Coefficients $c_i$ are the rates of the three reactions.

The correspondence between the two models relies in the fact that the trajectories of the Gillespie’s SSA tend to the solutions of the ODEs system given by the law of mass action on the reactions. This result is due to the particular application of the Kurtz’s theorem \[^13\] to chemical systems.

#### Lotka-Volterra GCPS Definition

The model above does not fill GCPS requirements since the first and last reactions are not pairwise interactions. We propose to extend reactions \[^3\] by considering a renewable resource $X$ for $Y_1$ as a third species \[^4\]. the molecular level of $X$ remains constant whatever is

\[^3\]We use the same notation as in \[^12\] to express that the food resource $X$ is assumed renewable.

89
Figure 1: Runs of the Lotka-Volterra model with renewable (left column) and not-renewable (right column) resources for different initial states (kinetics rates equal 1): $Y_1 = 500, Y_2 = 1500, X = X = 1000$ (first row), $Y_1 = Y_2 = X = X = 1000$ (second row), $Y_1 = Y_2 = 1000, X = X = 10000$ (third row), $Y_1 = 1000, Y_2 = 0, X = X = 10000$ (fourth row). The solid red line represents preys, the dashed green line predators, and the blue dotted line resources. The two first rows show that both dynamics exhibit the same properties as presented in [12] (particularly, in second row, oscillations raise from an equilibrium initial state for the ODEs). The third row shows the difference in the dynamics when the resource size is ten times larger than the population size. The last row shows the difference in the dynamics when the predator population is empty. The simulations have been done using the general simulation language MGS (http://mgs.spatial-computing.org) that allows an easy implementation of all models of the present article [20, 14, 21].
its production or its consumption. The extended system of reactions is:

\[ \begin{align*}
X + Y_1 & \xrightarrow{c'_1} 2Y_1 \\
Y_1 + Y_2 & \xrightarrow{c'_2} 2Y_2 \\
Y_2 + X & \xrightarrow{c'_3} 2X
\end{align*} \]  

(4)

The use of a pairwise interaction in the last reaction can be interpreted as a competition between the two predator behaviours: a predator in presence of preys eats and reproduces (second reaction); a predator in absence of prey (represented by the grass) dies (third reaction). Moreover, with the hypothesis that the number of X remains constant, the behaviour of this system is exactly described by ODEs \((2)\) with \(c_1 = Xc'_1, c_2 = c'_2\) and \(c_3 = Xc'_3\). Thus, systems \((3)\) and \((4)\) are equivalent in terms of dynamics.

System \((4)\) is only composed of pairwise interactions that satisfy condition 4 of Definition 1. Thus, it can be easily translated to a one-symbol GCPS, denoted \(\Pi_{LV}\), working in fs-mode (in concentration-dependent implementation) with rules \(R\):

\[ \begin{align*}
(\emptyset,0)(\emptyset,1) & \rightarrow (\emptyset,1)(\emptyset,1) \\
(\emptyset,1)(\emptyset,2) & \rightarrow (\emptyset,2)(\emptyset,2) \\
(\emptyset,2)(\emptyset,0) & \rightarrow (\emptyset,0)(\emptyset,0)
\end{align*} \]

where membrane indices 0, 1 and 2 represent the environment (an infinite source of \(X\)), the preys \(Y_1\) and the predators \(Y_2\), respectively.

Let now consider the previously defined concentration-dependent evolution with probabilities \(p_\mu = a_\mu / a_0\) for each \(\mu \in R\) with the propensity function \(a_\mu = c_\mu h_\mu\). \(c_\mu\) is the rate of the corresponding reaction in \((3)\) and \(h_\mu\) is given by equation \((1)\) accordingly to \(\Pi_{LV}\). The reader is invited to pay attention that even if the environment is an infinite source of \(X\) (instead of a constant one), the dynamics are well taken into account: rules involving the environment have probabilities that do not depend on the environment size, see equation \((1)\). For example, the propensity of the first reaction is given by \(a_1 = c_1 h_1 = c_1 Y_1 = c'_1 X Y_1\) as expected w.r.t. reactions \((4)\). In this respect, any computation of \(\Pi_{LV}\) represents a run of the Gillespie’s SSA of reactions \((4)\). As a consequence, \(\Pi_{LV}\) is an exact model of the original Lotka-Volterra system.

It has to be remarked that \(\Pi_{LV}\) cannot be described by any PP since the environment objects are involved in its definition. A possible specification of the Lotka-Volterra equations may be obtained within a PP by considering \(X\) as a not-renewable resource. Such a definition has been realized (taking reactions \((4)\) and substituting \(X\) by \(X\)). However, due to the limitation of resource, this system does not respect the dynamics of equation \((2)\) anymore. For example, without any predators, a population of preys stabilizes in this model, while in the original model it grows exponentially. Figure \(\ref{fig1}\) gives some examples of simulations of the Lotka-Volterra model considering renewable and not-renewable resources.

**General Population Dynamics.** It is possible to reverse the above method and to give a GCPS system whose population dynamics will correspond to some dynamics given by a system of ODEs, under the following conditions. Let us consider the ODEs system defined on set of variables \(\{Y_1,\ldots,Y_N\}\) of the form

\[ \frac{dY_i}{dt} = \sum_{j,k} a_{jk}^i Y_k - \sum_j (b_{ij} + b_{ji}) Y_i Y_j \]

(5)

where coefficients \(a_{jk}^i\) and \(b_{ij}\) satisfy the following conditions:

1. for all \(i, j, k\), \(a_{jk}^i \geq 0\) and \(b_{ij} \geq 0\);

2. for all \(j, k\) such that \(b_{jk} \neq 0\), there exists either one index \(i_0\) such that \(a_{jk}^{i_0} = 2b_{jk}\), or two distinct indices \(i_1\) and \(i_2\) such that \(a_{jk}^{i_1} = a_{jk}^{i_2} = b_{jk}\); for any other index \(i, i \neq i_0\) or \(i \neq i_1\) and \(i \neq i_2\), \(a_{jk}^i = 0\).
The above conditions are sufficient to ensure that $\sum_i \frac{dY_i}{dt} = 0$. Then there exists a concentration-dependent fs-mode GCPS without rules involving the environment (i.e., a PP) whose behaviour is exactly described by ODEs \(^5\) when the population size goes to the infinity. Indeed, these equations correspond to the mass-action law of a set of rules such that for any $j, k$ with $b_{jk} \neq 0$

\[(\bullet, j)(\bullet, k) \rightarrow^{b_{jk}} (\bullet, i_0)(\bullet, i_0)\text{ or } (\bullet, j)(\bullet, k) \rightarrow^{b_{jk}} (\bullet, i_1)(\bullet, i_2)\]

according to the considered possibility of the above condition 2. The reader is invited to pay attention that these equations correspond to a wider range of dynamics than the dynamics of second-order chemical reactions with two products since they allow the specification of ordered interactions (e.g., involving a sender and a receiver as considered in the PP literature). This property also holds in PP and suggests that equations \(^5\) exactly describe PP dynamics when the size of the populations tends to the infinity.

It is obvious that a more general class of population behaviours is captured by concentration-dependent fs-mode GCPS since they have not to be conservative thanks to the environment. Following the idea of equivalence between systems \(^3\) and \(^4\) in terms of dynamics, equations \(^5\) can be extended with the introduction of a renewable variable $\overline{Y}_0$. This wider class of ODEs is supported by concentration-dependent fs-mode GCPS model.

### 4 Computational Properties

In this section, we focus on the original use of PP as a computational model of algebraic numbers proposed in \(^5\). This article investigates the case where the computation is independent of the initial contents of the system.

Using the GCPS terminology, the main idea of \(^5\) is to consider the result of a computation as a ratio between the number of tokens in certain membrane and the total number of tokens (without taking care of the environment) when the population size goes to the infinity and when the state of the system converges. The proposed work relies on the definition of a particular strategy of execution of the PP: a step of execution consists in sampling uniformly and independently of the past two distinct tokens in the membrane and let them interact in a sequential mode. This strategy is fair since it corresponds to a Markov process. The authors of the aforementioned article studied the Markov chain associated with PP and proved its equivalence to some system of ODEs at the limit.

We remark that the same kind of result directly arises from considerations of Section 3.2 since this computational model is captured by one-symbol GCPS working in fs-mode with Gillespie concentration-dependent implementation. Indeed, the above execution strategy exactly corresponds to a Gillespie’s SSA run where the stochastic constants equal 1 for all rules. Thus, the study of the model corresponds to the investigation of the sensibility of the associated ODEs system. Let us illustrate this point by considering the running example of \(^5\)

\begin{align*}
(\bullet, p)(\bullet, p) &\rightarrow (\bullet, p)(\bullet, m) \\
(\bullet, p)(\bullet, m) &\rightarrow (\bullet, p)(\bullet, p) \\
(\bullet, m)(\bullet, p) &\rightarrow (\bullet, p)(\bullet, p) \\
(\bullet, m)(\bullet, m) &\rightarrow (\bullet, p)(\bullet, m)
\end{align*}

where symbols $p$ and $m$ identify two membranes. It has been shown that the ratio $\frac{p}{p+m}$, where $p$ (resp. $m$) is the size of the membrane $p$ (resp. $m$), converges to $\frac{1}{\sqrt{2}}$ when the population size goes to the infinity.
Accordingly to equations (5), we associate ODEs with this GCPS as follows

\[
\frac{dY_p}{dt} = Y_m^2 + 2Y_pY_m - Y_p^2 \\
\frac{dY_m}{dt} = -Y_m^2 - 2Y_pY_m + Y_p^2
\]

The stable states of this system are obtained when the two equations vanish, that is, when either \( Y_m = -\sqrt{2} + 1 \) or \( Y_m = \sqrt{2} - 1 \). The first solution is incoherent since it involves a negative size of population. The second solution trivially leads to the expected result \( Y_p \approx Y_m = \frac{1}{\sqrt{2}} \).

## 5 Conclusions

In this article we investigated connections between population protocols and generalized communicating P systems. The two models share the same multiset structure and the same type of rules. Traditionally PP are used to study population dynamics in the context of distributed algorithmics while GCPS are investigated for the computational properties.

By incorporating the derivation mode from PP into GCPS framework we obtained a strict inclusion of PP in GCPS working in fs-mode. We then took a particular implementation of the fs-mode corresponding to a run of the Gillespie’s SSA and we obtained that the dynamics of the systems can be described by the corresponding system of differential equations. Different questions then could be explored, like the investigation of the conditions ensuring that the system reaches a stable state regardless of its initial state or ensuring that a stable state is never reached for any initial configuration. GCPS are in this sense easier to handle than PP because of the environment that permits to easily simulate the equivalent of creation or degradation reactions. Section 5.2 also considers the converse problem of the construction of a GCPS system exhibiting a particular behaviour given by a systems of ODEs. It would be interesting to see if the given sufficient conditions are also necessary. A mathematical challenge resulting from Section 4 is whether for any algebraic number \( x \in [0..1] \) there is a GCPS working in concentration-dependent evolution implementation of the fs-mode that converges to \( x \).

We remark that the presented results hold only in the concentration-dependent implementation of the fs-mode. By taking an equiprobable implementation the results are completely different.

Since Petri Nets can be seen as multiset rewriting, it is clear that the results of this paper can be translated to this domain (for Petri Nets with specific type of rules and an additional fairness strategy).

We think that the fs-mode has interesting properties that should be further explored. As showed in the article, the fairness condition is in some sense similar to a stochastic evolution, so it could be preferable to consider this condition instead of a stochastic behaviour. Another interesting property of the proposed stochastic implementation is that Gillespie’s SSA introduces an explicit continuous time and discrete events in the model, which do not appear in a GCPS description.

**Acknowledgments.** The authors would like to acknowledge the support of ANR project SynBioTIC.

**References**

[1] D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta. Computation in Networks of Passively Mobile Finite-State Sensors. *Distributed Computing*, pages 235–253, Mar. 2006.

[2] D. Angluin, J. Aspnes, and D. Eisenstat. Stably Computable Predicates are Semilinear. In *PODC’06: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 292–299, New York, NY, USA, 2006. ACM Press.
[3] J. Aspnes and E. Ruppert. An Introduction to Population Protocols. *Bulletin of the Europ. Assoc. for Theor. Comp. Sci.*, 93:98–117, Oct. 2007.

[4] F. Bernardini, M. Gheorghe, M. Margenstern, and S. Verlan. How to Synchronize the Activity of All Components of a P System? *International Journal of Foundations of Computer Science.*, 19(5):1183–1198, 2008.

[5] O. Bournez, P. Chassaing, J. Cohen, L. Gerin, and X. Koegler. On the Convergence of Population Protocols when Population Goes to Infinity. *Applied Mathematics and Computation*, 215(4):1340–1350, 2009.

[6] G. Ciobanu, L. Pan, G. Paun, and M. J. Pérez-Jiménez. P Systems with Minimal Parallelism. *Theor. Comput. Sci.*, 378(1):117–130, 2007.

[7] E. Csuhaj-Varjú, G. Vaszil, and S. Verlan. On Generalized Communicating P Systems with One Symbol. In M. Gheorghe, T. Hinze, and G. Paun, editors, *Proceedings of the Eleventh International Conference on Membrane Computing*, pages 137–154. Verlag ProBusiness Berlin, 2010.

[8] E. Csuhaj-Varjú and S. Verlan. On Generalized Communicating P Systems with Minimal Interaction Rules. *Theor. Comp. Sci.*, 412(1-2):124–135, 2011.

[9] L. Edelstein-Keshet. *Mathematical Models in Biology*. Random House, New York, 1988.

[10] R. Freund and S. Verlan. A Formal Framework for Static (Tissue) P Systems. In G. Eleftherakis, P. Kefalas, G. Păun, G. Rozenberg, and A. Salomaa, editors, *Membrane Computing, 8th International Workshop, WMC 2007, Thessaloniki, Greece, June 25-28, 2007 Revised Selected and Invited Papers*, volume 4860 of LNCS, pages 271–284. Springer, 2007.

[11] P. Frisco. *Computing with Cells*. Oxford University Press, 2009.

[12] D. T. Gillespie. Exact Stochastic Simulation of Coupled Chemical Reactions. *J. Phys. Chem.*, 81(25):2340–2361, 1977.

[13] T. Kurtz. A Limit Theorem for Perturbed Operator Semigroups with Applications to Random Evolutions. *J. Funct. Anal.*, 12:55–67, 1973.

[14] O. Michel, A. Spicher, and J.-L. Giavitto. Rule-Based Programming for Integrative Biological modelling – Application to the modelling of the Lambda Phage Genetic Switch. *Natural Computing*, 8(4):865–889, December 2009.

[15] M. Presburger. Über die Vollstandigkeit eines gewissens Systems der Arithmetik Ganzer Zahlen, in Welchem die Addition als Einzige Operation Hervortritt. *Comptes rendus du I Congres des Mathematiciens des Pays Slaves*, pages 92–101, 1929.

[16] G. Păun. *Membrane Computing. An Introduction*. Springer–Verlag, 2002.

[17] G. Păun, G. Rozenberg, and A. Salomaa. The Oxford Handbook Of Membrane Computing. Oxford University Press, 2009.

[18] W. Reisig. *Petri Nets. An Introduction*. Springer, 1985.

[19] G. Rozenberg and A. Salomaa. *Handbook of Formal Languages, 3 volumes*. Springer, 1997.

[20] A. Spicher, O. Michel, M. Cieslak, J.-L. Giavitto, and P. Prusinkiewicz. Stochastic P Systems and the Simulation of Biochemical Processes with Dynamic Compartments. *BioSystems*, 91(3):458–472, March 2008.

[21] A. Spicher, O. Michel, and J.-L. Giavitto. *Understanding the Dynamics of Biological Systems*, chapter Interaction-Based Simulations for Integrative Spatial Systems Biology, 195–231. Springer, 2011.

[22] S. Verlan, F. Bernardini, M. Gheorghe, and M. Margenstern. Generalized Communicating P Systems. *Theor. Comp. Sci.*, 404(1-2):170–184, 2008.