Scientific Annals of Computer Science vol. 21, 2011, pp. 227–247

Generalized Communicating P Systems Working in Fair Sequential Mode

Antoine SPICHER[1], Sergey VERLAN[1]

Abstract

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 that permits to ensure a particular sequence of configurations. We show that PP can be seen as a particular variant of GCPS. We also consider several stochastic evolutions satisfying different fairness conditions and particularly focus on those corresponding 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.

Keywords: Population protocols, P systems, Stochastic simulation, Gillespie's algorithm.

1 Introduction

The notion of a *generalized communicating P system* was introduced in [25], 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 $(a, i)(b, j) \to (a, k)(b, l)$ where a and b are objects

[1] LACL, Département Informatique, Université Paris Est, 61 av. gén. de Gaulle, 94010, Créteil, France.
Email: {antoine.spicher,verlan}@u-pec.fr

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 [20] where tokens from various input places come along together to fire a given transition and then fork out to destination places, see [25, 5] for more details.

Depending on the form of the communication rules, several *restrictions on communication rules* 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 [25, 10, 9, 19] it was proved that eight of the possible nine restricted variants (modulo symmetry) 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 are able to achieve this generative power even with relatively small numbers of cells and simple underlying (hypergraph) architectures. In [9] a further restriction is introduced by considering that the alphabet of objects is a singleton (like in Petri Nets) and it is shown that 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, modeled 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 [6] 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 observe that the fairness condition is used to select particular evolutions among all possible runs. In a more general case, the fairness was investigated in the area of transition systems and there are several possible definitions that can be used to yield the desired behavior [7]. We consider such conditions 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 behavior of the system in this mode. There are several possible evolution strategies implementing different fairness conditions and we are particularly interested in the stochastic ones. We focus on the strategy corresponding 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 can be represented by a system of ODEs when the population size goes to the infinity. We also 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) behavior or computing approximations of algebraic numbers.

We remark that this paper is an extended version of [24].

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 [18, 19, 21].

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, λ. 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 $M : V \longrightarrow \mathbb{N}$; $M(a)$ 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 $x = a_1^{M(a_1)} a_2^{M(a_2)} \dots a_n^{M(a_n)} \in V^*$, 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 $\Sigma_{a \in V} |x|_a$. The difference and the sum of two multisets m_1 and m_2 are denoted as $m_1 - m_2$ and $m_1 + m_2$ respectively.

2.1 Population Protocols

A *population protocol* is the quintuple $\mathcal{P} = (Q, \Sigma, \imath, \omega, \delta)$, where Q and Σ are alphabets (elements of Q are called *states* and those of Σ *input symbols*), $\imath : \Sigma \to Q$ is the *initial state mapping*, $\omega : Q \to \{0, 1\}$ is the *individual* *output function* and $\delta \subseteq Q^4$ is a total transition function. Function δ can be described by using the notation $q_1q_2 \to q_1'q_2'$, for $(q_1, q_2, q_1', q_2') \in \delta$. By convention we consider that $q_1q_2 \to q_1q_2$ when no rule involving q_1q_2 in the left hand side is present.

The computation of a population protocol on the input multiset $w \in \Sigma^*$ is performed by n *agents* ($n = |w|$), each of them having a *state* that is an element of Q. Two agents being in states q_1 and q_2, respectively, may interact by the rule $q_1q_2 \to q_1'q_2'$ from δ. As a result of such an interaction the states of the agents change to q_1' and q_2' respectively.

More formally, a configuration of a PP is a multiset C over Q. A transition between two configurations C and C', written as $C \Rightarrow C'$, is obtained by a single interaction between the agents, *i.e.*, $C \Rightarrow C'$ if $C' = C - \{q_1, q_2\} + \{q_1', q_2'\}$ for some $q_1, q_2 \in C$ and $q_1q_2 \to q_1'q_2' \in \delta$. We remark that the transition does not change the size of the configuration.

An *execution* of the population protocol on the input multiset w is defined as an infinite sequence of configurations $C_0, C_1, C_2, \ldots$ obtained by a sequence of transitions $C_i \Rightarrow C_{i+1}$ for all $i \geq 0$, starting from the initial configuration $C_0 = \imath(w)$.

An execution is *fair* if for all configurations C that appear infinitely often in the execution, if $C \Rightarrow C'$ for some configuration C', then C' appears infinitely often in the execution.

The output value of configuration C, denoted by $\omega(C)$, is defined as follows:

$$\omega(C) = \begin{cases} 0, & \text{if } \omega(q) = 0, \text{ for all } q \in C, \\ 1, & \text{if } \omega(q) = 1, \text{ for all } q \in C, \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

Let p be a predicate over multisets of elements of Σ. Predicate p can be considered as a function whose range is $\{0, 1\}$ and whose domain is the collection of these multisets. The predicate p is said to be computed by a PP $\mathcal{P}$ as defined above if for any input multiset w and every fair execution of $\mathcal{P}$ on the input w, there exists a number $k \in \mathbb{N}$ such that $\omega(C_{k'}) = p(w)$ for all $k' > k$.

The following result was proved in [1, 2]:

Theorem 1 *A predicate is computable in the population protocol model iff it is semilinear.*

Recall that semilinear sets are known to correspond to predicates on counts of input agents definable in first-order Presburger arithmetic [17].

2.2 Generalized Communicating P Systems

Next we recall the basic definitions concerning generalized communicating P systems [25].

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* Π*;*
2. *$E \subseteq O$; called the* set of environmental objects *of* Π*;*
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, labeled 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, labeled 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 $(a,i)(b,j) \rightarrow (a,k)(b,l)$, 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 Π, 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* Π is the tuple $(\lambda, w_1, \ldots, w_n)$.

Given a multiset of rules $\mathcal{R}$ over R and a configuration $C = (z_0, z_1, \ldots, z_n)$ of Π, we say that $\mathcal{R}$ is *applicable* to C 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 $C = (z_0, z_1, \dots, z_n)$ of Π, a new configuration $C' = (z'_0, z'_1, \dots, z'_n)$ 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 $C' = (z'_0, z'_1, \dots, z'_n)$ 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 [8] the concept of the derivation mode had to be precisely defined and [12] 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 Π from configuration C to configuration C'. A transition sequence is said to be a *successful generation* by Π if it starts with the initial configuration of Π and ends with a *halting configuration* in which a *halting condition* is satisfied. Generally, the *total* halting condition is considered, being true for a configuration where no further transition step can be performed.

We say that Π *generates a non-negative integer* n if there is a successful generation by Π 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 Π 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 [25] 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 [25, 10, 9] 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 [13]. For more details on Petri Nets and membrane computing we also refer to [19].

In this article we shall consider the dynamics of the configuration of GCPS, so we are no more interested in computation (and halting evolutions).

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 (fs-mode) for GCPS and hence we are able to encode any PP in a GCPS w.r.t. their dynamics. In some conditions, an equivalence between the two models can be stated.

We consider several fairness conditions for GCPS and their implementations with stochastic and Gillespie-like strategies.

3.1 GCPS in Fair Sequential Mode

For GCPS, mainly the maximally parallel derivation mode is investigated with several attempts to investigate asynchronous or minimally parallel derivation mode, see [19] 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 corresponding to the definition 3 below. We can consider such a strategy in GCPS case as well.

The notion of *fairness* was originally developed in the context of transition systems to select a particular subset of all the possible executions of a system. There exist many ways of defining fairness conditions. We give below three different fairness conditions widely found in the literature [7].

Definition 2 *A sequence of transitions $C_0 \Rightarrow C_1 \Rightarrow \ldots$ is* strong locally fair *(SLF) iff for every rule r, if r is applicable in infinitely many configurations C_i, then there are infinitely many j such that one passes from C_j to C_{j+1} ($C_j \Rightarrow C_{j+1}$) by applying rule r.*

Definition 3 *A sequence of transitions $C_0 \Rightarrow C_1 \Rightarrow \dots$ is* strong globally fair *(SGF) iff for every C and C' such that $C \Rightarrow C'$, if $C = C_i$ for infinitely many i, then $C_j = C'$ for infinitely many j.*

Definition 4 *A sequence of transitions $C_0 \Rightarrow C_1 \Rightarrow \dots$ is* weakly fair *(WF) iff for every rule r, if it exists j such that r is applicable in configuration C_i for all $i \geq j$, then for infinitely many k one passes from C_k to C_{k+1} ($C_k \Rightarrow C_{k+1}$) by applying rule r.*

When assuming that the number of configurations is finite (this is the case for classical PP for example), the definition 3 can be easily rephrased as follows: a computation $C_0 \Rightarrow C_1 \Rightarrow \dots$ is fair if

- there exists a non-negative integer N such that configuration C_N belongs to a terminal strongly connected component of the state graph[1]; and
- any state of this terminal strongly connected component appears infinitely often in the execution.

In the context of this paper, we are interested in the study of the behavior of GCPS in sequential mode with some fairness condition: we say that a GCPS Π evolve in a *fair sequential* derivation mode (fs-mode) w.r.t. to the fairness condition $\mathcal{F}$ if the sequence of transitions between configurations of Π fulfills the fairness condition $\mathcal{F}$.

Encoding PP in GCPS. 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 $\mathcal{P}$ with set of states Q (for convenience we suppose that $Q = \{1, \dots, n\}$) and transition relation δ in an initial configuration C_0, let define GCPS $\Pi_\mathcal{P} = (O, E, w_1, \dots, w_n, R, 1)$ as follows

- $O = E = \{\bullet\}$,

[1] In this directed graph, nodes correspond to the configurations C, and two nodes C and C' are directly linked if $C \Rightarrow C'$.

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

The above system encodes each state q of $\mathcal{P}$ by a token $\bullet$ present in membrane labeled by q. It is easy to see that there is a one-to-one correspondence between the sequences of transitions of $\mathcal{P}$ and of $\Pi_{\mathcal{P}}$.

Conversely, given a GCPS Π with an empty environmental set (*i.e.*, $E = \emptyset$) and with a deterministic rule set R (eventually extended to a total function with the identity), it is possible to define PP $\mathcal{P}_\Pi$ with $Q = O \times \{1, \ldots, n\}$ and $\delta = R$, *i.e.*, each couple (object, cell) corresponds to a state in $\mathcal{P}_\Pi$. The initial configuration $C_0 \in Q^*$ of $\mathcal{P}_\Pi$ is given such that $|C_0|_{(o,i)} = |w_i|_o$, for all $(o, i) \in Q$.

Such an encoding works only if the set E is empty, otherwise it is not possible to represent an infinite multiset (corresponding to the environment) using PP.

From these considerations, it is trivial to establish that $PP \equiv GCPS$ in fs-mode with respect to SGF when $E = \emptyset$ and R is deterministic.

If we consider an encoding function $\imath$ as defined for PP and the halting condition corresponding to the stabilization of the ω-image of the configuration, then as an immediate consequence of [1, 2], we obtain that any GCPS working in fs-mode with respect to SGF and that has a deterministic rule set not 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) is computationally complete if the number agents in some particular state q_0 is unbounded (going to the infinity).

3.2 Simulation of GCPS in FS-Mode

When simulating a transition system the notion of fairness should be replaced by a local strategy that guarantees the validity of the condition.

Simulating a GCPS in fs-mode consists in defining an algorithm, called a *scheduler*, that decides at each time step which applicable rule of the current state has to be applied. A scheduler can be seen as a local strategy that guarantees the validity of the fairness condition during the whole execution. It is characterized by the subset of sequences of transitions it can generate and then by the fairness condition it respects.

Among all possible implementations of schedulers, Markovian processes feature prominently since they do not require any history or global knowledge on the state space, and they provide a modeling tool useful in many domains (like in the simulation of population behaviors or in distributed algorithmics). Such a Markovian process corresponds to a labeling of the state graph arrows $C \Rightarrow^{p_r(C)} C'$ by a static probability $p_r(C)$ that only depends on configuration C and on the applied rule $r = (\bullet, q_1)(\bullet, q_2) \to (\bullet, q_1')(\bullet, q_2')$ (without loss of generality we can focus on one token GCPS). We give here two examples of Markovian processes:

1. *Equiprobable implementation*: $C \Rightarrow^{1/k_C} C'$ where k_C denotes the cardinality of the set $\{C' \,|\, C \Rightarrow C'\}$.

2. *Concentration-dependent implementation*: $C \Rightarrow^{p_r(C)} C'$ where $p_r(C)$ is proportional to $h_r(C)$, the number of distinct combinations of tokens in C that activate r and to a stochastic constant c_r that only depends on r. The number $h_r(C)$ is given by

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

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

It is possible to give sufficient conditions for a Markov process to generate fair sequences of transitions with probability 1 with respect to some fairness condition. For example, Theorem (1) from [4] establishes that any *bounded fair transition system*, that is a transition system where there exists a real number c with $0 < c < 1$ such that $p_r \geq c$ for all transitions $C \Rightarrow^{p_r} C'$, generates fair sequences of transitions with probability 1 w.r.t. SLF. This result can be straightforwardly applied for the equiprobable implementation: since any GCPS has sets of rules R of finite size, we have $1/k_C \geq 1/|R|$ for any configuration C. Therefore, the equiprobable implementation generates fair sequences of transitions with probability 1 w.r.t. SLF.

To our knowledge, there is no such general result applicable for the concentration-dependent implementation. Nevertheless, for the class of GCPS equivalent to PP (see Section 3.1), a *random pairing scheduler* (consisting in choosing an ordered pair of agents at random, independently and uniformly from all pairs) generates fair sequences of transitions with probability 1 for SGF [1]. The proof of this result relies on the fact that a PP has a finite state graph and that the associated Markov chain leaves non-terminal strongly connected components and visits infinitely often all states of one of the terminal strongly connected components with probability 1. Following the same reasoning, this result also holds for any concentration-dependent scheduler on GCPS with an empty environment. In fact, the usual PP random pairing scheduler is a particular concentration-dependent scheduler where $p_r(C) = \frac{h_r(C)}{\sum_{s \in R} h_s(C)}$.

3.3 Gillespie's SSA

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 [14] a *stochastic simulation algorithm* (SSA) for producing the trajectories of such a chemical system by computing the *next reaction* and the *elapsed time* since last reaction occurred. In the following, we show that Gillpesie's SSA provides a concentration-dependent scheduler where the transition probabilities depend on a stochastic coefficient associated with each rule.

Specification of the SSA. Let μ be a chemical reaction. The probability that μ takes place during an infinitesimal time step is proportional to:

- c_μ, the *stochastic reaction constant*[2] of reaction μ;
- $h_\mu(S)$, the number of distinct molecular combinations that can activate reaction μ; 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 μ and will occur in the time interval $(t+\tau, t+\tau+d\tau)$ is:

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

The function $a_\mu(S) = c_\mu\, h_\mu(S)$ is called the *propensity* of reaction μ, and $a(S) = \sum_\nu a_\nu(S)$ is the combined propensity of all reactions.

This probability leads to the first straightforward Gillespie's *exact* stochastic simulation algorithm called the *first reaction method.* From a

[2] Evaluating the stochastic constants is one of the key issues in stochastic simulations of biochemical reactions.

chemical state S, it consists in choosing an elapsed time τ for each reaction μ 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.

Gillespie's Simulation of a GCPS. Gillespie's SSA can be used in wider range of contexts than only (bio)chemical modeling. Applied to a GCPS Π, SSA gives a way to simulate a continuous-time Markov chain with the states corresponding to the configurations of Π and with transitions between states corresponding to a single application of a rule of Π. From this point of view, the definition of a GCPS is extended to associate with each rule $r:(a,i)(b,j)\rightarrow^{c_r}(a,l)(b,k)$ a *stochastic coefficient* c_r (*i.e.*, a positive real number) that characterizes the kinetics of the rule in the Markov chain. The probability for a transition between two configurations C and C' corresponding to the application of a rule r with stochastic coefficient c_r is given by

$$p_r(C)=\frac{a_r(C)}{a(C)}=\frac{c_r h_r(C)}{a(C)}$$

that directly corresponds to the definition of a concentration-dependent scheduler.

As an immediate consequence, Gillespie's algorithm generates fair executions of PP w.r.t. SGF; in other words, it allows the computation of semilinear predicates with probability 1.

The use of Gillespie's SSA allows us to deal with interesting population behaviors in GCPS. It is illustrated in the two next sections where the Gillespie's simulation of GCPS is used to characterized population dynamics.

4 Application to the Lotka-Volterra Model

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

4.1 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 [11]. This model specifies how two coupled populations (of chemicals or individuals) Y_1 (the preys) and Y_2 (the predators) behave. It may be presented using the following ODEs

$$\frac{dY_1}{dt} = (c_1 - c_2 Y_2)Y_1 \qquad \frac{dY_2}{dt} = (c_2 Y_1 - c_3)Y_2 \tag{2}$$

In [14], D.T. Gillespie proposes the study of this model from a discrete and stochastic point view based on the system of chemical reactions

$$Y_1 \xrightarrow{c_1} 2\,Y_1 \qquad Y_1 + Y_2 \xrightarrow{c_2} 2\,Y_2 \qquad Y_2 \xrightarrow{c_3} . \tag{3}$$

The dynamics of these reactions is conveniently characterized using the predator-prey interpretation. 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 on the fact that the trajectories of Gillespie's SSA tend to the solutions of the ODEs system given by the *law of mass action* on the reactions when the number of chemicals tends to the infinity (see Section 5). This result is due to the particular application of Kurtz's theorem [15] to chemical systems.

4.2 Lotka-Volterra GCPS Definition

The model above does not fulfill the GCPS requirements since the first and last reactions are not pairwise interactions. We propose to extend reactions (3) by considering a *renewable* resource $\overline{X}$ for Y_1 as a third species[3]: the molecular level of X remains *constant* whatever its production or its consumption. The extended system of reactions is:

$$\overline{X} + Y_1 \xrightarrow{c'_1} 2\,Y_1 \qquad Y_1 + Y_2 \xrightarrow{c'_2} 2\,Y_2 \qquad Y_2 + \overline{X} \xrightarrow{c'_3} 2\,\overline{X} \tag{4}$$

The use of a pairwise interaction in the last reaction can be interpreted as a competition between the two predator behaviors: 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

[3] We use the same notation as in [14] to express that the food resource X is assumed renewable.

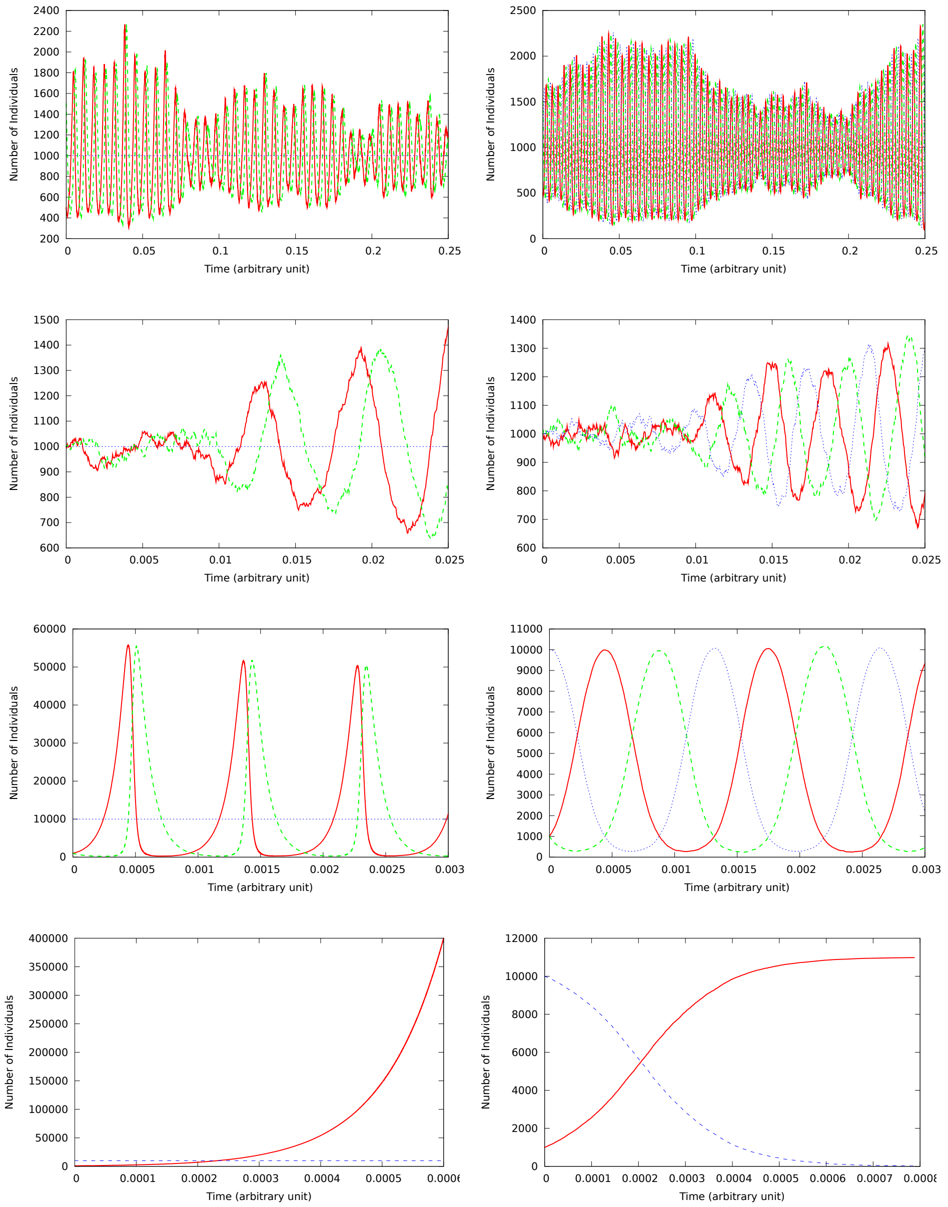

Figure 1: Runs of the Lotka-Volterra model with renewable (left column) and non-renewable (right column) resources for different initial states (kinetics rates equal 1): $Y_1 = 500, Y_2 = 1500, \overline{X} = X = 1000$ (first row), $Y_1 = Y_2 = \overline{X} = X = 1000$ (second row), $Y_1 = Y_2 = 1000, \overline{X} = X = 10000$ (third row), $Y_1 = 1000, Y_2 = 0, \overline{X} = X = 10000$ (fourth row). The solid red line represents preys, the dashed green line predators, and the dotted blue line resources. The two first rows show that both dynamics exhibit the same properties as presented in [14] (particularly, in second row, oscillations araise 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 [22, 16, 23].

hypothesis that the number of $\overline{X}$ remains constant, systems (3) and (4) have equivalent dynamics in terms of propensity when $c_1 = \overline{X}c'_1$, $c_2 = c'_2$ and $c_3 = \overline{X}c'_3$: propensity of the first reaction of system (3) is given by $a_1 = c_1\, h_1 = c_1\, Y_1 = c'_1\, \overline{X}\, Y_1$, that is the expected propensity of the first reaction of system (4); second reactions of both systems are the same; a similar reasoning can be done for the last case.

System (4) is only composed of pairwise interactions that satisfy condition 4 of Definition 1. Thus, it can be easily expressed as a one-symbol GCPS, denoted Π_{LV}, with rules R:

$$\begin{array}{ll}
(\bullet,0)(\bullet,1) \rightarrow^{c'_1} (\bullet,1)(\bullet,1) & \quad (\bullet,1)(\bullet,0) \rightarrow^{c'_1} (\bullet,1)(\bullet,1) \\
(\bullet,1)(\bullet,2) \rightarrow^{c'_2} (\bullet,2)(\bullet,2) & \quad (\bullet,2)(\bullet,1) \rightarrow^{c'_2} (\bullet,2)(\bullet,2) \\
(\bullet,2)(\bullet,0) \rightarrow^{c'_3} (\bullet,0)(\bullet,0) & \quad (\bullet,0)(\bullet,2) \rightarrow^{c'_3} (\bullet,0)(\bullet,0)
\end{array}$$

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 scheduler. A transition $C \Rightarrow^{p_\mu} C'$ has a probability of $p_\mu = a_\mu / a_0$ with the propensity function $a_\mu = c_\mu\, h_\mu$: c_μ is the rate of the corresponding reaction in (3) and h_μ is given by equation (1) accordingly to Π_{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 is well taken into account: rules involving environmental objects have probabilities that do not depend on the number of these objects, see equation (1). In this respect, any computation of Π_{LV} represents a run of the Gillespie's SSA of reactions (4). As a consequence, Π_{LV} is an exact model of the original Lotka-Volterra system.

It has to be remarked that Π_{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 *non-renewable* resource. Such a definition has been realized (taking reactions (4) and substituting $\overline{X}$ by X.) However, due to the limitation of resource, this system does not respect the original Lotka-Volterra system dynamics anymore. For example, without any predators, a population of preys stabilizes in this model, while in the original model it grows exponentially. Figure 1 gives some examples of simulations of the Lotka-Volterra model considering renewable and non-renewable resources.

5 GCPS With Infinite Population Size

In this section, we study how the dynamics of GCPS with infinite population can be described by a system of ODEs.

5.1 GCPS Population Dynamics

As previously said in the presentation of the Lotka-Volterra model, the *mass action law* gives a way to associate a system of ODEs with a system of chemical reactions. It is possible to reverse this method and to give a GCPS system whose population dynamics (when the population size tends to the infinity) will correspond to some dynamics given by a system of ODEs.

Let us consider the ODEs system defined on set of variables $\{Y_1, \ldots, Y_N\}$ of the form

$$\frac{dY_i}{dt} = \sum_{j,k} a^i_{jk} Y_j Y_k - \sum_j (b_{ij} + b_{ji}) Y_i Y_j \tag{5}$$

where coefficients a^i_{jk} and b_{ij} satisfy the following conditions:

1. for all i, j, k, $a^i_{jk} \geq 0$ 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 $a^{i_0}_{jk} = 2\, b_{jk}$, or two distinct indices i_1 and i_2 such that $a^{i_1}_{jk} = a^{i_2}_{jk} = b_{jk}$; for any other index i, $i \neq i_0$ or $(i \neq i_1$ and $i \neq i_2)$, $a^i_{jk} = 0$.

The above conditions are sufficient to ensure that $\sum_i \frac{dY_i}{dt} = 0$. Then there exists a concentration-dependent implementation of fs-mode GCPS with respect to the fairness condition SGF and without rules involving the environment (*i.e.*, a PP) whose behavior 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 $b_{jk} \neq 0$

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

according to the considered possibility of the above condition 2. The reader is invited to pay attention to the fact 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 behaviors is captured by concentration-dependent fs-mode GCPS since they do not need 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 $\overline{Y_0}$.

5.2 Computational Properties

We focus here on the original use of PP as a computational model of algebraic numbers proposed in [6]. 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 [6] 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. 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 5.1 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 [6]

$$\begin{aligned}(\bullet,p)(\bullet,p) &\to (\bullet,p)(\bullet,m)\\ (\bullet,p)(\bullet,m) &\to (\bullet,p)(\bullet,p)\\ (\bullet,m)(\bullet,p) &\to (\bullet,p)(\bullet,p)\\ (\bullet,m)(\bullet,m) &\to (\bullet,p)(\bullet,m)\end{aligned}$$

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 infinity. Accordingly to equations (5), we associate ODEs with this GCPS as follows

$$\frac{dY_p}{dt} = Y_m^2 + 2\,Y_pY_m - Y_p^2 \qquad\qquad \frac{dY_m}{dt} = -Y_m^2 - 2\,Y_pY_m + Y_p^2$$

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

Using this method, we found the 17 algebraic numbers computable with the 81 PP involving only two states: $\{0, 2-\sqrt{3}, \frac{3-\sqrt{5}}{2}, \frac{1}{3}, \frac{5-\sqrt{17}}{2}, \sqrt{2}-1, \frac{1}{2}, \frac{\sqrt{17}-3}{2}, \frac{1}{\sqrt{3}}, \frac{\sqrt{5}-1}{2}, 1-\frac{1}{\sqrt{2}}, 1-\frac{1}{\sqrt{3}}, 2-\sqrt{2}, \frac{2}{3}, \frac{1}{\sqrt{2}}, \sqrt{3}-1, 1\}$. In the general case, there are $\binom{n+1}{2}^{n^2}$ PP with n states and we do not know how many algebraic numbers are computable by them. We conjecture that only solutions of the system of equations $\frac{dY_i}{dt} = 0$ derived from the ODEs system 5 can be computed.

6 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 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 considered different fairness conditions and we concentrated on a particular implementation of the SGF corresponding to a run of Gillespie's SSA. As a result we obtained that the dynamics of such systems can be described by the corresponding system of differential equations. Different questions could then 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 5.1 also considers the converse problem of the construction of a GCPS system exhibiting a particular behavior 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 5.2 is whether for any algebraic number $x \in [0..1]$ being a root of a system of polynomials of degree 2 there is a concentration-dependent evolution implementation of GCPS working in the fs-mode with respect to SGF that converges to x.

We remark that the presented results hold only in the concentration-dependent scheduler. By taking an equiprobable scheduler the results are 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 types of rules and an additional fairness strategy).

We think that the fs-mode has interesting properties that should be further explored. The fairness condition can express a local stochastic evolution, so it could be preferable to consider this condition instead of a stochastic behavior. 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.

Acknowledgements

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*, 235–253, 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*, 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] C. Baier, M. Kwiatkowska. On the verification of qualitative properties of probabilistic processes under fairness constraints, *Information Processing Letters*, **66**, 71–79, 1998.

[5] 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.

[6] 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.

[7] J. Clement, C. Delporte-Gallet, H. Fauconnier, M. Sighireanu. Guidelines for the Verification of Population Protocols Proceedings of *31st International Conference on Distributed Computing Systems (ICDCS 2011)*, 215–224.

[8] G. Ciobanu, L. Pan, G. Paun, and M. J. Pérez-Jiménez. P systems with minimal parallelism. *Theor. Comput. Sci.*, 378(1):117–130, 2007.

[9] 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*, 137–154. Verlag ProBusiness Berlin, 2010.

[10] E. Csuhaj-Varjú and S. Verlan. On generalized communicating P systems with minimal interaction rules. *Theor. Comp. Sci.*, 412(1-2):124–135, 2011.

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

[12] 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*, 271–284. Springer, 2007.

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

[14] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81(25):2340–2361, 1977.

[15] T.G. Kurtz. The Relationships between Stochastic and Deterministic Models for Chemical Reactions. *Journal of Chemical Physics*, 57(7):2976–2978, 1971.

[16] O. Michel, A. Spicher, and J.-L. Giavitto. Rule-based programming for integrative biological modeling – application to the modeling of the lambda phage genetic switch. *Natural Computing*, 8(4):865–889, december 2009.

[17] M. Presburger. Uber die vollstandig-keit eines gewissen systems der arithmetik ganzer zahlen, in welchemdie addition als einzige operation hervortritt. *Comptes rendus du I Congres des Mathematicians des Pays Slaves*, pages 92–101, 1929.

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

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

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

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

[22] 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.

[23] 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.

[24] A. Spicher, S. Verlan. Generalized Communicating P Systems Working in Fair Sequential Mode. In G. Ciobanu, editor, *5th Workshop on Membrane Computing and Biologically Inspired Process Calculi*, 83–94, also arXiv:1108.3432v1 [cs.DC], 2011.

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