Parallel Graph Rewriting Systems

Dragoş Sburlan

Ovidius University
Faculty of Mathematics and Informatics
Constantza, Romania
dsburlan@univ-ovidius.ro

1 Introduction

Nowadays an increasing interest regards the study of the development of biological systems in which more species of individuals interact (usually to perform a certain global task). Research ranging from completely different areas like the study of metapopulations (the study of groups of spatially separated populations of the same species which live in fragmented habitats and interact at some level) and HIV infections was done in essentially the same manner. Traditionally, such studies were done by employing continuous models where (partial) differential equations were used to capture the dynamics of these systems.

Currently, the usage of discrete models where the system dynamics is captured from the collective actions of individual entities has been shown to be a promising choice. This is based on the fact that living organisms are spatially discrete and the individuals occupy particular localities at a given time. The interactions between individuals are strongly connected with their neighborhood relations.

While characterizing these facts a basic issue regards the way the space is represented. Simple models that involve no detailed spatial structure are in general analytically easily solvable. However, as the complexity of the reaction-diffusion dynamics grows, the models based on partial differential equations become intractable to be analyzed.

On the other hand, integrating within the model a detailed spatial structure (as cellular automata models do, for instance) the setback comes in general from the impossibility to analyze the models except only by performing simulations. Although such models have much greater biological reality, they suffer from the difficulty of generalization (hence of finding the exact behavior). This is especially important while formulating some practical testable predictions regarding a given model.

P systems are formal computing devices that were initially inspired and abstracted from the cell functioning (see [4]). In general, P systems make use of multisets to represent the computational support. These multisets are placed in-

side the membranes which in their turn are disposed in some hierarchical tree structure. The (maximal) parallel applications of some multiset rewriting rules (particular to each membrane) were used to process the multisets.

Although these formal systems were extensively studied with respect to their computational power and efficiency, while representing some biological processes many difficulties arise. Representing the data support as multisets essential simplifies the structure of the environment and of the individuals from within (the neighboring relations between the individuals are completely ignored), the focus being over the system dynamics. However, in this case, two main assumptions are considered: the environment is homogeneous so that the concentration of the individuals do not change with respect to space and the number of individuals of each species in the environment is "adequately" large (hence the concentration of the individuals might be assumed to vary continuously over the time). Moreover, the rules that describe the interactions between the individuals are assumed to be executed in a maximal parallel manner and governed by a global clock that marks equal steps.

Even if all these simplifications are useful while defining a computing formal framework, they are questionable if the aim is to model and simulate actual biological systems. This is way many new features that are meaningful to biologists were added to the original paradigm in order to extend its functionality and versatility for modeling.

In order to cope with these issues, probabilistic/stochastic P systems were introduced (see [2], [6], [1]). In general, the main idea was to associate to the rules some weights describing how they should be applied at a given moment. For a particular rule, the weight gives the susceptibility of its execution at certain instant. Hence, applying this principle to all interaction rules it sets up more realistic bounds of the nondeterministic application of the rules. The ultimate goal of this approach is to integrate the structural and dynamical characteristics of a real biosystem into the way the rules of the model are selected to be applied and executed (preserving at the same time the unstructured computing support). Although this method has in general good simulation time complexity it is inadequate if the interacting species are poorly represented, when there exist many "inactive" individuals (that are not the subject of any rule) with respect to the entire population of individuals, or when the environment is not homogeneous.

2 Preliminaries

We assume the reader familiar with the basic notions of P systems (one can consult [4] for more details), so that here we only recall some notions regarding the abstract rewriting systems on multisets. ARM systems represent a variant of P systems which was proposed in order to perform simulations of some bio-chemical processes. Later on, due to its modeling flexibility, it was used to study some symbiotic mechanism of an ecological system and even for proposing a novel theory of evolution.

ARMS is a stochastic model that uses multisets to represent the bio-chemical support. Multiset rewriting rules are used to describe the bio-chemical reactions. As opposed to the classical definition of P systems where the rules are applied in a nondeterministic, maximal parallel manner and with competition on the objects composing the multisets, in ARMS the rules obey the Mass Action Law where the frequency of a reaction follows the concentration of bio-chemicals and a rate constant. Consequently, the rules to be applied are chosen probabilistically from the rules set and each probability is given by the ratio of the total number of colliding chemicals of a reaction to the sum of the total number of colliding chemicals of every reactions in the rule; the applications of the rules remain parallel and with competition on the objects.

More formally, an ARM system is a construct $\Pi = (O, w, R)$ where O is the alphabet of objects, w represents the multiset of objects at the beginning of computation, and R is a set of multiset rewriting rules of type $u \xrightarrow{k} v$, where $u \in O^+$, $v \in O^*$, and $k \in \mathbb{R}$ is the rate constant of the rule.

For example, in case of a cooperative rule of type $r_i : aA + bB \rightarrow cC + dD$ and a given multiset of objects M, the probability of rule execution is defined as $Prob(r_i) = \frac{k_i M_A^a M_B^b}{R}$, where k is the rate constant (determined experimentally) and R is a coefficient for normalizing the probabilities ($\sum_i Prob(r_i) = 1$). Similarly, probabilities can be defined for any type of rules.

The system Π starts to evolve from the initial configuration (represented by w) by applying the rules in parallel, randomly selecting the rules but according with the probabilities computed as above. Π is governed by an universal clock that marks equal time units.

We have run more tests using an ARM system Π where $O = \{A, B, C, D, X, F\}$, and the set of rules R is given bellow:

$$
\begin{array}{ll}
r_1 : AB \xrightarrow{k_1} X & r_8 : F \xrightarrow{k_8} F \\
r_2 : AC \xrightarrow{k_2} X & r_9 : A \xrightarrow{k_9} A \\
r_3 : BD \xrightarrow{k_3} X & r_{10} : B \xrightarrow{k_{10}} B \\
r_4 : CD \xrightarrow{k_4} F & r_{11} : C \xrightarrow{k_{11}} C \\
r_5 : FX \xrightarrow{k_5} FF & r_{12} : D \xrightarrow{k_{12}} D \\
r_6 : FA \xrightarrow{k_6} FF & r_{13} : X \xrightarrow{k_{13}} X \\
r_7 : FB \xrightarrow{k_7} FF &
\end{array}
$$

The initial configuration of Π was $w = A^{250}B^{250}C^5D^5$ and in our tests we used several values for k_i, $1 \leq i \leq 13$. The system attempts to simulate the behavior of some interacting individuals, represented here as the objects A, B, C, and D, sharing the same environment. In addition, the individuals corresponding to the objects C and D (which are much less than the individuals corresponding to the objects A and B) share a localized patch in the environment. Thus, we assumed the environment not to be homogeneous.

If at least once the objects C and D interact (i.e., the rule $CD \xrightarrow{k_4} F$ is applied) they will produce an object F which will trigger the conversion of all existing objects in the multiset into F (the rules r_5, r_6, and r_7). The rest of the rules (r_8 till r_{13}) are used to slow down the rate of parallelism.

Since we have assumed the existence of a patch in the environment of individuals corresponding to objects C and D, then we could make another further assumption: if the patch is "large enough" so that there exists at least two individuals C and D which are not interacting initially with the individuals A and B, then there exists a "significant" probability that the rule $CD \xrightarrow{k_4} F$ is executed. While using multisets to represent the individuals in the environment we lose the structure, hence when simulating such systems we actually have to relay on the probabilities of the executions of the rules (which in their turn depend on some constants experimentally determined). In Figure 1, one can notice the different behaviors of the same system and they are related to the usage of such probabilities. The charts shown on the right hand side present a simulation when the rule $CD \xrightarrow{k_4} F$ was executed at least once, while the charts on the right hand side present a simulation when the rule $CD \xrightarrow{k_4} F$ was not executed at all. Although the model considered is very simple a similar situation might happen when representing some complex systems. Even more, such situations might emerge during the system evolution and sudden shifts in the behavior might arise from some minor changes in the circumstances; if this is the case, then it would make almost impossible the precise identification of the rate constants associated to the rules.

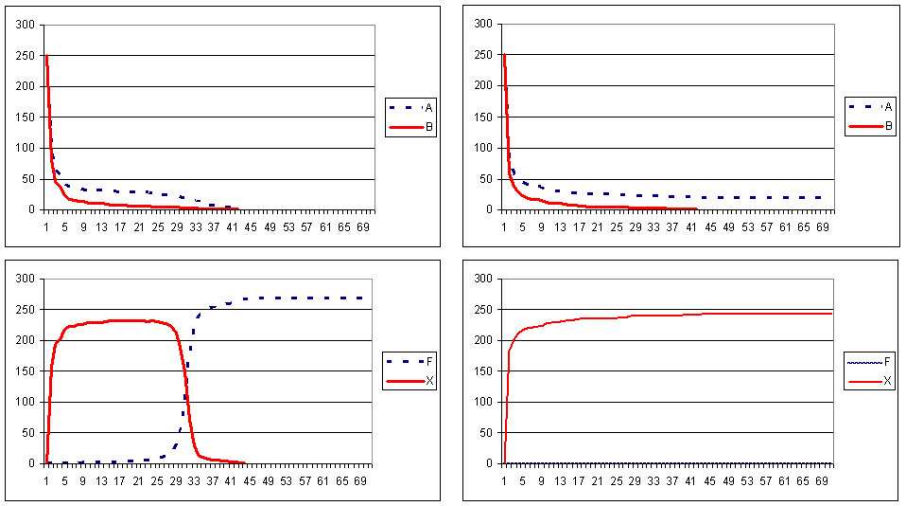

Fig. 1. Two runs of system Π. The results are presented on columns and they show the different behaviors of the same system when some minor changes in the circumstances happen.

Besides all of these issues, if the number of objects in the model decreases under a certain limit, the usage of probabilities to specify the way the rules are applied becomes inadequate.

3 PGR Systems

Aiming to tackle the mentioned issues, in this section we introduce a new model for simulating bio-systems composed by interacting individuals of various species in a given environment.

Denote by C the set of species in an environment represented here as a metric space (for simplicity, let $\mathbb{R}^k, k \geq 2$, be the environment). Let $V \subseteq L \times C$ be the finite set of labeled individuals in the environment (L denotes a finite set of labels that uniquely identify the individuals in the environment). In addition, let $f : V \to \mathbb{R}^k$, $k \geq 2$, be a bijective mapping; for a node $v = (n, l) \in V$, the value $h(v)$ denotes the position of the individual v in the environment. In addition let $r > 0$, $r \in \mathbb{R}$, be a positive constant.

Based on above definition one can represent the environment and the individuals from within as a graph $G_0 = (V_0, E_0)$ where $V_0 = V$ and the set of edges is constructed as follows: for two nodes $v_1, v_2 \in V$, if $h(v_1)$ belongs to the open ball centered in $h(v_2)$ and with radius r (i.e., $h(v_1) \in B(h(v_2), r)$) then there exists an edge from v_1 to v_2.

For simplicity we assume that G_0 is connected, that is, for any two nodes $m, n \in V$ there exists a sequence $m = v_0, v_1, \ldots, v_t = n \in V$ such that $h(v_i) \in B(h(v_{i-1}), r)$, for $1 \leq i \leq t$.

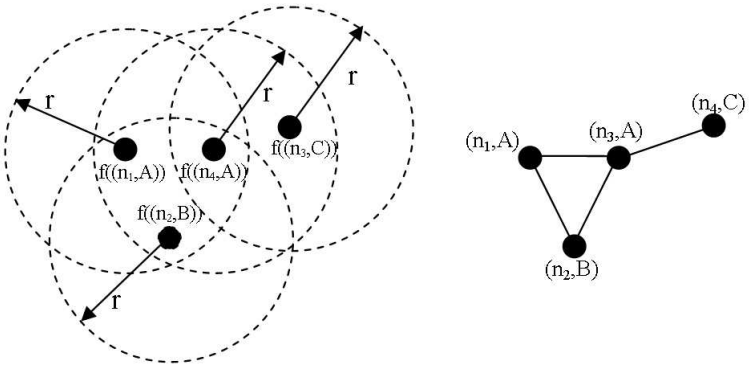

Fig. 2.

Motivated by these facts we can introduce the following model. A *parallel graph rewriting system* (in short, a PGR system) is a construct $\Gamma = (C, G_0, R)$ where

• $C = \{c_1, \ldots, c_k\}$ is a finite set of symbols;

• $G_0 = (V_0, E_0)$ is the *initial global graph* – a connected graph such that $V_0 \subseteq L \times C$ is a set of labeled nodes and $E_0 \subseteq V_0 \times V_0$ is a set of edges between nodes from V_0;

• R is a finite set of *graph rewriting rules.*

A graph rewriting rule $r \in R$ is of the following type:

$$r = (G_1 = (V_1, E_1), G_2 = (V_2, E_2)),$$

where $V_i \subseteq L_i \times C$, $E_i \subseteq V_i \times V_i$, $i \in \{1, 2\}$. The graphs G_1 and G_2 are connected graphs; G_1 represents the neighboring relations between the individuals that are required for an interaction to take place and G_2 represents the output of an actual interaction between individuals represented in G_1. In addition we will assume that G_1 and G_2 are not arbitrary graphs, but rather they obey some physical constraints: any node from G_1 and G_2 cannot be the subject of more than a constant $t_r \in \mathbb{N}$ edges – a condition that assume the nonexistence of more than t_r individuals in an open ball of radius r.

A graph rewriting rule $r = (G_1, G_2) \in R$ can be applied on a graph G if G_1 is *label isomorphic* with some subgraph $G_s = (V_s, E_s)$ of G, that is, there exists a bijective mapping $h : V_1 \to V_s$ such that $h((m, c)) = (n, c)$ and $h^{-1}((n, c)) = (m, c)$, where $(m, c) \in V_1$, $(n, c) \in V_s$ and such that any two nodes $u, v \in V_s$ are adjacent in G_s if and only if $h(u)$ and $h(v)$ are adjacent in G_1 (see Figure 3).

In other words, a graph rewriting rule r can be applied on G iff the left-hand side rule's graph is "contained" in G both as layout and as corresponding node labels (via an edge/label-preserving bijection).

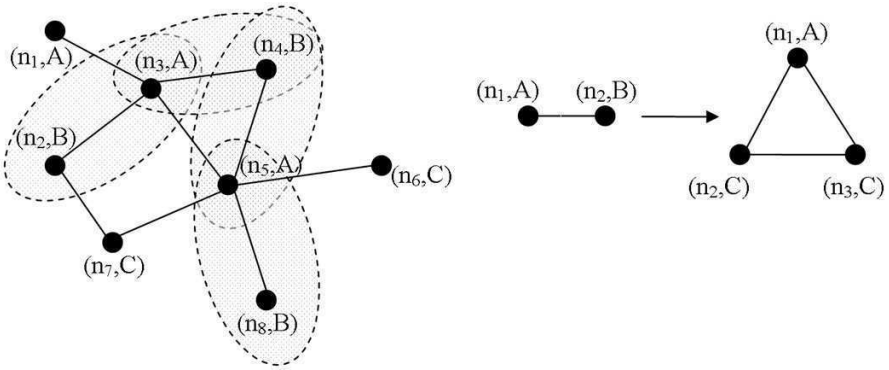

Fig. 3. A graph $G = (V, E)$ denoting the computing support and a graph rewriting rule $r = (G_1, G_2)$. The sites where the rule r can be applied in G are explicitly figured. If $G_s = (V_s = \{(n_4, B), (n_5, A)\}, E_s = \{((n_4, B), (n_5, A))\})$ then G_1 is label isomorphic with G_s. The neighborhood set of degree $k = 1$ of G_s is $B_1 = \{(n_3, A), (n_6, C), (n_7, C), (n_8, B)\}$.

Applying a rule r over G follows the following steps:

- eliminate G_s from G (all the nodes from V_s are eliminated from V; all the edges of the type (v, v_s), $v \in V$, $v_s \in V_s$ are deleted from E);
- add G_2 to G (some relabeling of the nodes from G_2 is required in order to avoid duplicates of nodes at multiple application of r). All the (relabeled) nodes and edges of G_2 are added to G;
- add a set of edges from some nodes of V_2 to some nodes of $V \setminus (V_s \cup V_2)$. The edges are established as described below.

For the graph G_s let us define the *neighborhood set* of degree k

$$B_k = \{v \in V \setminus V_s \mid \text{ there exists a path of length less or equal with } k \text{ from } v \text{ to a node } u \in V_s\}.$$

As we mentioned above, the output of the application of a rule consists of new individuals that, by hypothesis, at the moment of their apparition it is assumed to belong to the same vicinity. How big is that vicinity and how the new individuals are related to the rest depend on many factors among which we just mention the type of the rule and the environment. Consequently, in our framework, the set B_k is useful when defining the new neighborhood relations triggered by the application of a rule. By some straightforward physical arguments, the output graph G_2 of the rule r is likely to be "connected" to G via the nodes from B_k. However, for simplicity, we will consider the neighborhood set of degree 1 in our simulations.

Let $\overline{E} = \{(n_1, n_2) \in E \mid n_1 \in B_1, n_2 \in V_s\}$. Then, a number equals with $card(\overline{E})$ of random edges from the nodes of G_2 to the nodes from B_1 are added to G but such that any node considered is not the subject of more than t_r edges.

Starting from the initial configuration (the initial global graph G_0), the system evolves according to the rules from R and the current labeled graph in a non-deterministic parallel manner (but not necessarily maximal). The labeled graph of Γ at any given moment constitutes the configuration of the system at that moment. For two configurations G_A and G_B we can define a transition from G_A to G_B if we can pass from G_A to G_B by applying rules from R.

Determining whether two graphs are isomorphic is referred to as the *graph isomorphism problem*. Although this problem belong to **NP** it is neither known to be solvable in polynomial time nor it is **NP**-complete. A generalization of this problem (that is used in our formalism) is the subgraph isomorphism problem which is **NP**-complete; hence the known deterministic algorithms for this problem are exponential.

Remark 1. There is a physical motivation to assume that after applying a rule of the system, the newly produced objects (that correspond to the output nodes of the rule) belongs to the same vicinity, hence the left hand side graph of any rule should be complete.

Remark 2. For a given PGR system, as much as the radius r grows (hence the number of edges in the initial global graph is close to $\frac{n(n-1)}{2}$ where n is the number of the nodes, that is, the initial global graph is "almost" complete) and the degrees

of the neighboring sets grow as well, the result of a simulation is similar with one obtained using parallel multiset rewriting. This is because multisets can be seen in our formalism as complete graphs, hence any individual in the system is in a neighboring relation with any other individual (hence, they can interact if proper rules exist).

4 PGRS Simulator and a Test Case

The simulator implements the model introduced in Section 3. Its main characteristics regard the definition of the rules set by using an XML file, and the possibility to save/load intermediate configurations. The simulator is written in Java language hence it benefits of cross-platform compatibility, parallelism, and possibility to distribute the computational effort.

The task that has the most computational resource consumption is the subgraph isomorphism problem which is addressed whenever a rule $r = (G_1, G_2)$ is selected for application and the set S of all the subgraphs of the global graph that are isomorphic with G_1 has to be determined. Even more, whenever a subgraph $\overline{G} \in S$ is selected to be rewritten by r, a run through all the elements of S has to be performed in order to eliminate those subgraphs that have some nodes from $\overline{G}$. Considering all these matters for all the rules from the rule set and a relatively small global graph, the overall time complexity for simulating just one computational step is exponential. Nevertheless, if the left hand side graphs of the rules from the rule set are very simple (i.e., less than 4 nodes) and the global graph contains at most hundreds of nodes, the problem is feasible. Moreover, taking into account that the problem can be easily parallelized one can divide the the problem into smaller instances and distribute them over a network.

Let us consider the following PGR system $\Gamma = (C, G_0, R)$ where
- $C = \{A, B, C, D, F, X\}$,
- $R = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8\}$ is defined as follows:

In our tests, the initial global graph G_0 was build to obeying some properties. First of all, a random graph G_0' was generated and this graph contains 500 nodes labeled only with A and B (the apparition of these labels are equally probable) and 2000 edges. A second graph G_0'' was generated and this graph contains 10 nodes labeled only with C and D (the apparition of these labels are equally probable) and 30 edges. Finally, G_0' and G_0'' were merged together in order to form G_0 by connecting 10 randomly chosen nodes from G_0' with 10 randomly chosen nodes from G_0''.

We ran the simulator for 100 times, considering for each run a new initial global graph generated as above. In Figure 4 are represented the minimal and the maximal values at each step of the simulation for the objects A, B, C, and D. Any particular simulation graphic from our test case lay between the boundaries established.

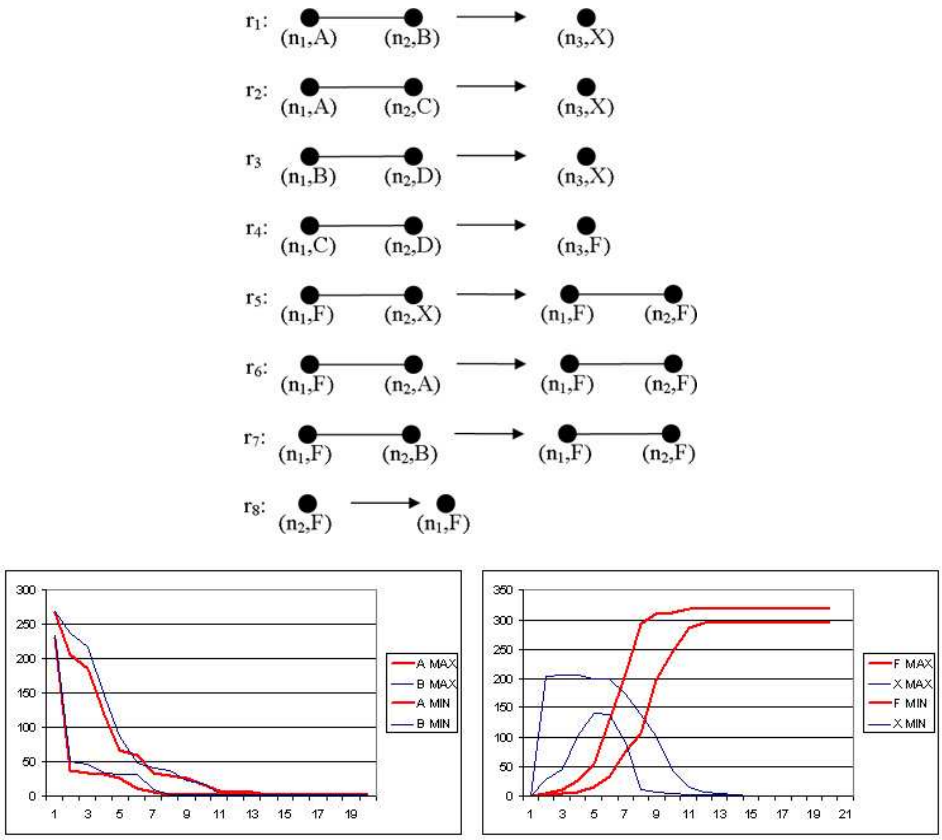

Fig. 4. The results of 100 simulations of different GPR systems but having the same properties. The minimal and maximal obtained values are explicitly marked.

5 Conclusions

Simulations performed using PGR systems in some cases give more accurate answers than ARMS simulations because they explicitly use the spatial distribution of individuals (hence the neighborhood relations can be extensively expressed). However the price to pay while using PGR systems regards the computational effort which in their case is exponential as time complexity. Nevertheless, for some cases when the number of interacting individuals in the environment is small and they are not dense, the PGR systems might be useful for performing simulations.

In order to handle these issues, a hybrid system combining features from the ARM and PGR systems might be proposed. Two directions could be taking into account:

- one can use alternatively an ARMS-type simulation whenever the number of individuals from all the species is large and a PGRS-type simulation whenever

the number of individuals from certain species goes below some threshold; in this case the newly obtained system uses in a more careful manner the probabilities for the rules executions.

- one can use in parallel an ARMS-type simulation over a multiset of many individuals and a PGRS-type simulation on relatively small instances of graphs. Then one can consider a time sequence and at each moment in the sequence one can merge the ARMS configuration with the multiset of labels of the nodes from the graph (or one can exchange some data between these simulations). In this way, the newly obtained hybrid systems become more robust against some unexpected changes in the behavior (which might be triggered by some minor changes).

References

1. D. Besozzi, P. Cazzaniga, D. Pescini, G. Mauri: Modelling metapopulations with stochastic membrane systems. *BioSystems*, 91 (2008), 499–514.
2. M. Cavaliere, I.I. Ardelean: Modeling biological processes by using a probabilistic P system software. *Natural Computing*, 2, 2 (2003), 173–197.
3. D.T. Gillespie: Exact simulation of coupled chemical reactions. *J. Physical Chemistry*, 81 (1977), 2340–2361.
4. Gh. Păun: *Membrane Computing. An Introduction.* Springer, Berlin, 2002.
5. D. Pescini, D. Besozzi, G. Mauri, C. Zandron: Dynamical probabilistic P systems. *Int. J. Found. Comput. Sci.*, 17, 1 (2006), 183–204.
6. Y. Suzuki, H. Tanaka: Modelling p53 Signaling Pathways by Using Multiset Processing. *Applications of Membrane Computing* (G. Ciobanu, Gh. Păun, M. Pérez-Jiménez, eds.), Springer, Berlin, 2006.
7. Y. Suzuki, J. Takabayashi, H. Tanaka: Investigation of a tritrophic system in ecological systems by using an artifcial chemistry. *J. Artif. Life Robot.*, 6 (2002), 129–132.
8. Y. Suzuki, S. Tsumoto, H. Tanaka: Analysis of cycles in symbolic chemical system based on abstract rewriting systems on multisets. *Proc. Intern. Conf. on Artificial Life 5* (Alife 5), 1996, 482–489.