HYBRID NUMERICAL P SYSTEMS

Ana Brânduşa Pavel[1] and Ioan Dumitrache[2]

We propose a new class of membrane systems, hybrid numerical P Systems. It has been previously shown that numerical P systems and their extension, enzymatic numerical P systems, are universal computing models. Programs can be written by defining a set of rules and a compartmental structure inspired by the transport mechanisms of the cell. The rules and the transport between membranes were originally designed as discrete components. In this paper, we present a novel class of numerical P systems in which we integrate continuous rules into the discrete cellular mechanisms of membrane computing paradigm. Therefore, we define hybrid numerical P systems, which incorporate both continuous production functions and discrete enzymatic guard conditions. We propose a novel, parallel and flexible modeling framework for dynamical hybrid systems.

Keywords: Membrane Computing, Numerical P Systems, Enzymatic Numerical P Systems, Hybrid Numerical P Systems, Dynamical Systems, Systems Biology

1. Introduction

P systems represent a computational paradigm inspired by the cell architecture and functioning. Several classes of P systems have been introduced in the framework of membrane computing [14].

Numerical P systems (NP systems) are a type of P systems, inspired by the cell's structure, in which numerical variables evolve inside the compartments by means of programs; a program (or rule) is composed of a production function and a repartition protocol. The variables have a given initial value and the production function is a multivariate polynomial. The value of the production function for the current instance is distributed among variables in certain compartments according to a repartition protocol. The formal definition of NP systems can be found in [13] where the authors introduce this type of P systems with applications in economics.

[1] Department of Automatic Control and Systems Engineering, Politehnica University of Bucharest, Splaiul Independenţei 313, 060042 Bucharest, Romania, e-mail: anabrandusa@gmail.com

[2] Department of Automatic Control and Systems Engineering, Politehnica University of Bucharest, Splaiul Independenţei 313, 060042 Bucharest, Romania, e-mail: idumitrache@ics.pub.ro

NP systems were designed both as deterministic and non-deterministic systems [13], [2]. An extension of NP systems, Enzymatic Numerical P systems (ENP systems), in which enzyme-like variables allow the existence of more than one rule in each membrane, while keeping the deterministic nature of the system, were introduced in [8]. Due to their properties, ENP systems represent a more powerful modeling tool for robot controllers than classical NP systems [9], [10], [16]. Previous work has shown that NP and ENP systems are universal computing models [13], [15]. Moreover, both sequential [16] and parallel [7] simulators for ENP systems have been developed.

In this paper we propose a new class of ENP systems, hybrid numerical P systems (HNP systems). A HNP system is an ENP system with continuous time rules. In order to model the differential equations of a dynamical system, we modify the mechanism by which the rules are applied and the production functions are distributed to the variables. The rules model kinetic laws and dynamical equations. We also present a case study of a biological dynamical system represented by this new class of P systems.

The authors are aware of other modeling frameworks for dynamical systems. Systems Biology Markup Language (SBML) is an XML based language used in systems biology to define dynamical systems. Also, hybrid Petri nets and hybrid automata are other broadly used modeling paradigms for hybrid systems. For example, the Alur-Henzinger hybrid automaton was developed primarily for algorithmic analysis of hybrid systems model checking [4, 1, 6]. However, the main advantages of HNP systems are their compartmental and naturally parallel structure. We propose HNP systems as a general modeling framework for large scale applications of hybrid systems. Such framework can be used to simulate the evolution of dynamical systems such as cells and tissues growth, microscopic organisms interaction, regulation of biochemical pathways, complex ecological systems and cloud computing applications etc.

2. **Numerical P systems (NP systems)**

Theoretical elements of membrane computing research area are presented in detail in [11], [12], [14]. Next, we formally introduce numerical P systems and their enzymatic version.

In order to define numerical P systems we need to define the following concepts. The basic one is the cell-like *membrane structure*, with the membranes labeled in a one-to-one manner with elements of an alphabet H. In its compartments, we have *variables*; those from region i are written in the form $x_{j,i}, j \geq 1$. The value of $x_{j,i}$ at time $t \in \mathbf{N}$ is denoted by $x_{j,i}(t)$.

In order to evolve the values of variables, we use *programs (rules)*, composed of two components, a *production function* and a *repartition protocol*. The former can be any function with variables from a given region. Using such a function we compute a *production value* of the region at a given time, depending on the values of variables at that time. This value is distributed

to variables from the region where the rule resides, and to variables in its upper and lower compartments (for a given region i, let $v_1, \ldots, v_{n_i}$ be all these variables) according to the repartition protocol associated with the used production function. The repartition protocols are of the following form:

$$c_1|v_1 + c_2|v_2 + \cdots + c_{n_i}|v_{n_i},$$

where $c_1, \ldots, c_{n_i}$ are natural numbers. The notation $c_i|v_i$ suggests that coefficient c_i corresponds to variable v_i in the repartition protocol. The idea is that coefficients $c_1, \ldots, c_{n_i}$ specify the proportion of the current production value which is distributed to each variable $v_1, \ldots, v_{n_i}$.

Formally, for a rule:

$$F_{l,i}(x_{1,i}, \ldots, x_{k_i,i}) \to c_{l,1}|v_1 + c_{l,2}|v_2 + \cdots + c_{l,n_i}|v_{n_i}$$

let,

$$C_{l,i} = \sum_{s=1}^{n_i} c_{l,s}; \quad q = \frac{F_{l,i}(x_{1,i}(t), \ldots, x_{k_i,i}(t))}{C_{l,i}}.$$

At a time $t \geq 0$ we compute $F_{l,i}(x_{1,i}(t), \ldots, x_{k_i,i}(t))$. The value q represents the "unitary portion" to be distributed to variables $v_1, \ldots, v_{n_i}$ proportionally with $c_{l,1}, \ldots, c_{l,n_i}$. Thus, $v_{l,s}$ will receive $q \cdot c_{l,s}, 1 \leq s \leq n_i$. The variables involved in the production function are reset to zero after computing the production; a variable not involved in a production function retains its value. After repartition, the quantities assigned to each variable from the repartition protocol are added to the current value of these variables (starting with 0 for the variables which were reset by a production function).

Thus, a *numerical P system* is a construct of the form:

$$\Pi = (m, H, \mu, (Var_1, Pr_1, Var_1(0)), \ldots, (Var_m, Pr_m, Var_m(0)), x_{j_0,i_0}),$$

where m is the degree of the system (the number of membranes), H is an alphabet of labels for membranes in μ, μ is a membrane structure with m membranes labeled injectively by elements of H, Var_i is the set of variables from region i, Pr_i is the set of rules from region i (all sets Var_i, Pr_i are finite), $Var_i(0)$ is the vector of initial values for the variables in region i, and x_{j_0,i_0} is a distinguished variable (from a distinguished region i_0), which provides the result of a computation.

Each rule is of the form specified above: $pr_{l,i} = (F_{l,i}(x_{1,i}, \ldots, x_{k_i,i}) \to c_{l,1}|v_1 + c_{l,2}|v_2 + \cdots + c_{l,n_i}|v_{n_i})$ denotes the l-th rule from region i, where the set $Var_i = \{x_{1,i}, \ldots, x_{k_i,i}\}$ and $v_1, \ldots, v_{n_i}$ are all variables from region i, the upper region and the immediately inner regions.

Such a system evolves in the way informally described before. Initially, the variables have the values specified by $Var_i(0), 1 \leq i \leq m$. A transition from a configuration at time instant t to a configuration at time instant $t+1$ is made by (i) choosing non-deterministically one rule from each region, (ii) computing the value of the respective production function for the values of local variables at time t, and then (iii) computing the values of variables at

time $t+1$ as indicated by repartition protocols. A sequence of such transitions forms a computation, with which we associate a set of numbers, namely, those which occur as values of the variable x_{j_0,i_0}; we consider only the positive values of x_{j_0,i_0}, and their set is denoted by $N^+(\Pi)$.

3. Enzymatic numerical P systems (ENP systems)

An ENP system is defined as an NP system with special enzyme-like variables which control the execution of the rules, as following:

$$\Pi_{ENP} = (m, H, \mu, (Var_1, Pr_1, Var_1(0)), \ldots, (Var_m, Pr_m, Var_m(0)), v_{j_0,i_0})$$

where:

- m is the degree of the membrane system (the number of membranes), $m \geq 1$;
- H is an alphabet of membrane labels;
- μ represents the tree structure of the membrane system;
- v_{j_0,i_0} is a distinguished variable from a compartment i which provides the results of the computation;
- Each membrane is defined by a 3-tuple:
  (1) Var_i is a (finite) set of variables from compartment i;
  (2) $Var_i(0)$ are the initial values of the variables from compartment i;
  (3) Pr_i is the set of rules from compartment i. Rules have one of the two following forms:
    (a) non-enzymatic form, which functions as like the rules in standard NP systems

$$Pr_{j,i} = (F_{j,i}(x_{1,i}, \ldots, x_{k_i,i}), c_{j,1}|v_1 + ... + c_{j,n_i}|v_{n_i})$$

    (b) enzymatic form

$$Pr_{j,i} = (F_{j,i}(x_{1,i}, \ldots, x_{k_i,i}), e_{t,i}, c_{j,1}|v_1 + ... + c_{j,n_i}|v_{n_i})$$

    where $e_{t,i}$ is an enzyme-like variable which controls the activation of the rule.

Like in NP systems, a rule is composed of a production function, a repartition protocol and optionally an enzyme-like variable. Each rule is evaluated in three steps, activation-production-distribution. First of all, it is established which rules are active. There can be more than one active rule in a membrane or none. A rule is active if it is in the non-enzymatic form or if the associated enzyme has a greater value than one of the variables involved in the production function. All active rules in the membrane system are executed in parallel in one computational step.

4. Hybrid numerical P systems (HNP systems)

A HNP system is a ENP system in which the rules model the transfer rates of differential equations. A production function will now have two components, the expression of the function and its transfer rate. The production

function multiplied with the transfer rate will be distributed to the variables in the repartition protocol, being multiplied with the corresponding repartition coefficients. After the rule is applied, the variables in the production function will not be consumed entirely as in classical NP systems. The production function value multiplied with the transfer rate will be subtracted from the value of each variable in the production function. Then the differential equation rule of each variable will be computed by summing the produced (positive) or consumed (negative) values of that variable from all rules in which it is present. The enzymatic mechanism will function like it was previously defined in the ENP systems. Therefore, we combine continuous time rules with enzyme-like conditions which act as guards controlling the programs flow.

A HNP system is formally defined as following:

$$\Pi_{HNP} = (m, H, \mu, (Var_1, Pr_1, Var_1(0)), \ldots, (Var_m, Pr_m, Var_m(0)), v_{j_0,i_0})$$

where:

- m is the degree of the membrane system (the number of membranes), $m \geq 1$;
- H is an alphabet of membrane labels;
- μ represents the tree structure of the membrane system;
- v_{j_0,i_0} is a distinguished variable from a compartment i which provides the results of the computation;
- Each membrane is defined by a 3-tuple:
  (1) Var_i is a (finite) set of variables from compartment i;
  (2) $Var_i(0)$ are the initial values of the variables from compartment i;
  (3) Pr_i is the set of rules from compartment i. Rules have one of the two following forms:
    (a) non-enzymatic form

$$Pr_{j,i} = (F_{j,i}(x_{1,i}, \ldots, x_{k_i,i}), K_{j,i}(y_{1,i}, \ldots, y_{q_i,i}), c_{j,1}|v_1 + ... + c_{j,n_i}|v_{n_i})$$

    (b) enzymatic form

$$Pr_{j,i} = (F_{j,i}(x_{1,i}, \ldots, x_{k_i,i}), K_{j,i}(y_{1,i}, \ldots, y_{q_i,i}), e_{t,i}, c_{j,1}|v_1 + ... + c_{j,n_i}|v_{n_i})$$

where $e_{t,i}$ is an enzyme-like variable which controls the activation of the rule.

For example a simple rule, such as $A \xrightarrow{K} 1|B+2|C$, produces the following system of equations:

$$\begin{aligned} \frac{dA}{dt} &= -K \cdot A \\ \frac{dB}{dt} &= K \cdot A \\ \frac{dC}{dt} &= 2 \cdot K \cdot A \end{aligned}$$

In case we want to limit the amount of A that can be transformed, we can add an enzymatic condition: $A|_e \overset{K}{\to} 1|B + 2|C$. This means that the rule is executed if and only if $min\{A\} < e \Leftrightarrow A < e$.

5. Case study of a biological dynamical systems modeled using HNP systems

We will further present a biological dynamical system modeled using HNP systems. We choose as an example the model proposed by V.A. Kuznetsov and M.A. Taylor [5], which describes the competition between the tumor and immune cells. The authors assume that tumor-immune interactions can be described by Michaelis-Menten equations. The model describes the response of the immune system's effector cells (ECs) to the growth of tumor cells (TCs). The penetration of a TC by an EC forms an EC-TC complex that can either produce the death of TCs or inactivate the ECs. Figure 1 shows the kinetic scheme of the interactions between ECs and TCs. E, T, C, E* and T* represent the concentrations of ECs, TCs, EC-TC complexes, inactivated ECs, and inactivated TCs. k_1 and k_1 are the rates of bindings of ECs to TCs and the detachment of ECs from TCs without deactivating the TCs. k2 is the rate at which EC-TC interactions produce the death of TCs. k3 is the rate at which EC-TC interactions inactivate ECs.

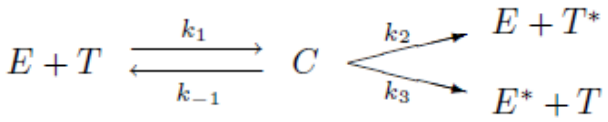


FIGURE 1. Kinetic laws which describe the interactions between ECs and TCs (figure from [3])

The equations describing the model are the following:

$$\begin{aligned}
\frac{dE}{dt} &= s + F(C,T) - d_1 \cdot E - k_1 \cdot E \cdot T + (k_{-1} + k_2) \cdot C \\
\frac{dT}{dt} &= a \cdot T \cdot (1 - b \cdot T) - k_1 \cdot E \cdot T + (k_{-1} + k_3) \cdot C \\
\frac{dC}{dt} &= k_1 \cdot E \cdot T - (k_{-1} + k_2 + k_3) \cdot C \\
\frac{dE^*}{dt} &= k_3 \cdot C - d_2 \cdot E^* \\
\frac{dT^*}{dt} &= k_2 \cdot C - d_3 \cdot T^*
\end{aligned}$$

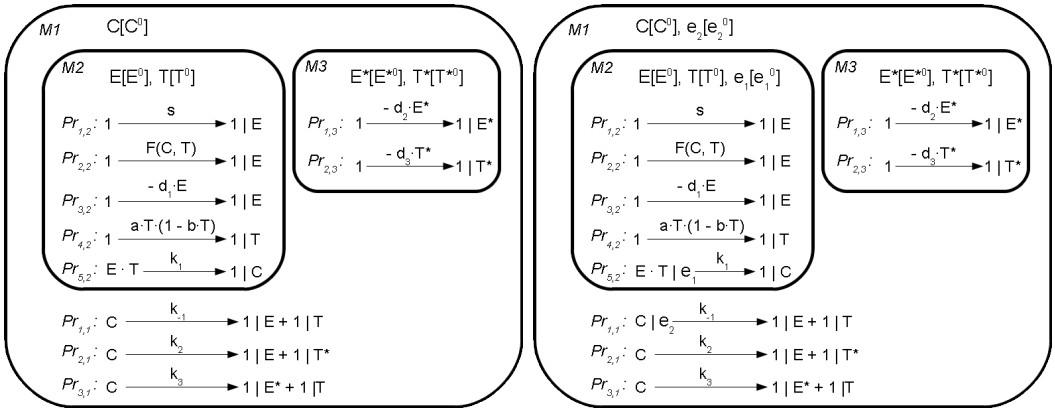
where,

- s is the normal rate of adult ECs into the tumor site (i.e., not increased by the presence of the tumor);
- F(C, T) is the accumulation of ECs in the tumor site; [5] assumes the following form for function F: $F(C,T) = F(E,T) = \frac{p \cdot E \cdot T}{r+T}$;
- d_1, d_2, and d_3 are the coefficients of the processes of inactivation and migration of E, E* and T*;
- a is the coefficient of maximal growth of the tumor;
- b is the environment's capacity.

Figure 2a illustrates the membrane system which is equivalent to Kuznetsov and Taylor dynamical system [5]. To be noticed that the kinetic laws from figure 1 are naturally described by the rules of membrane M_1 and $Pr_{5,2}$ of membrane M_2. The differential equation law of each variable, X, is obtained by accumulation of the production functions which generate X, P_j, multiplied with the transfer rates of the rules (shown above arrows in figure 2a), K_j, and the corresponding repartition coefficients, $c_{j,X}$.

Figure 2b presents a similar HNP systems to the one from figure 2a. It shows how enzyme-like variables can be incorporated in the structure, similarly to ENP systems. The enzymes act as guards, controlling which rules are active, based on the enzymatic conditions:

- $e_1 > min\{E^0, T^0\}$, in $Pr_{5,2}$;
- $e_2 > C^0$, in $Pr_{1,1}$.



(A) HNP system which describes Kuznetsov and Taylor model [5]

(B) HNP system with enzyme-like variables

FIGURE 2. HNP systems

6. **Conclusions**

In this paper, we propose a novel modeling framework for hybrid systems using membrane computing paradigm. HNP systems represent a general framework for modeling and simulation of dynamical systems with applications

in systems biology, systems engineering, robotics, etc. We show the expressiveness and flexibility of HNP systems by modeling a biological dynamical system. Future work includes applying the framework for more complex models in systems biology, ecology and robotics and evaluating the advantages compared to other modeling frameworks for hybrid systems.

REFERENCES

[1] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P. H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.

[2] C. Buiu, C. I. Vasile, and O. Arsene. Development of Membrane Controllers for Mobile Robots. *Information Sciences*, 187:33–51, March 2012. doi: 10.1016/j.ins.2011.10.007.

[3] M. Galach. Dynamics of the Tumor - Immune System Competition - the Effect of Time Delay. *Int. J. Appl. Math. Comput. Sci.*, 13(3):395–406, 2003.

[4] T.A. Henzinger. The Theory of Hybrid Automata. *Proceedings of the Eleventh Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 278–292, 1996.

[5] V.A. Kuznetsov and M.A. Taylor. Nonlinear dynamics of immunogenic tumors: Parameter estimation and global bifurcation analysis. *Bull. Math. Biol.*, 56(2):295–321, 1994.

[6] N. Lynch, R. Segala, and F. Vaandrager. Hybrid I/O Automata. *Information and Computation*, 185(1):103–157, 2003.

[7] M.A. Martínez-del Amor, M. García-Quismondo, L.F. Macías-Ramos, L. Valencia-Cabrera, A. Riscos-Núñez, and M.J. Pérez-Jiménez. Simulating p systems on gpu devices: A survey. *Fundamenta Informaticae*, 136:269–284, 2015.

[8] A. B. Pavel, O. Arsene, and C. Buiu. Enzymatic Numerical P Systems - a New Class of Membrane Computing Systems. In *The IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2010) Liverpool*, pages 1331–1336, September 2010.

[9] A. B. Pavel and C. Buiu. Using Enzymatic Numerical P Systems for Modeling Mobile Robot Controllers. *Natural Computing*, 11(3):387–393, 2012.

[10] A. B. Pavel, C. I. Vasile, and I. Dumitrache. Robot Localization Implemented with Enzymatic Numerical P Systems. In *Living Machines 2012: The International Conference on Biomimetic and Biohybrid Systems*, Proceedings in Lecture Notes in Artificial Intelligence, Barcelona, Spain, July 2012. Springer.

[11] Gh. Păun. Computing with Membranes. *Journal of Computer and System Sciences*, 61:108–143, 2000.

[12] Gh. Păun. *Membrane Computing - An Introduction.* Springer-Verlag, Berlin, 2002.

[13] Gh. Păun and R. Păun. Membrane Computing and Economics: Numerical P Systems. *Fundamenta Informaticae*, pages 213–227, 2004.

[14] Gh. Păun, G. Rozenberg, and A. Salomaa, editors. *The Oxford Handbook of Membrane Computing.* Oxford University Press, 2010.

[15] C. I. Vasile, A. B. Pavel, I. Dumitrache, and Gh. Păun. On the Power of Enzymatic Numerical P Systems. *Acta Informatica*, 49(6):395–412, 2012.

[16] C. I. Vasile, A. B. Pavel, J. Kelemen, and I. Dumitrache. Implementing obstacle avoidance and follower behaviors on Koala robots using Numerical P Systems. In *Proceedings of the 10th Brainstorming Week on Membrane Computing*, volume 2, pages 215–228, Seville, Spain, January-February 2012.