**MOKSLAS – LIETUVOS ATEITIS**
**SCIENCE – FUTURE OF LITHUANIA**
ISSN 2029-2341 / eISSN 2029-2252
*http://www.mla.vgtu.lt*

ELEKTRONIKA IR ELEKTROTECHNIKA
ELECTRONICS AND ELECTRICAL ENGINEERING

2015 7(3): 285–290

Sistemų inžinerija, kompiuterių technologija T 120
System Engineering, Computer Technologies T 120

http://dx.doi.org/10.3846/mla.2015.784

# APPLICATION PROSPECTS OF METABOLIC P SYSTEM

**Darius KULAKOVSKIS**
*Vilnius Gediminas Technical University, Vilnius, Lithuania*
*E-mail darius.kulakovskis@vgtu.lt*

**Abstract.** Metabolic P (MP) systems are a class of P systems and are based on metabolic process which is used by living organisms. We review for the purpose of experimental research the MP system development, implementation and modeling. To implement MP systems in dedicated hardware, some specifications need to be determined. We look into the possibility of modeling MP systems in fixed point arithmetic capable hardware and determine the minimal needed bit count for selected MP system model. A way to improve the accuracy of fixed point calculations is proposed, which includes the modification of MP grammar. This modification is done using a simple rule to multiply and divide constants in MP grammar equations. The resulting calculations have more than 250 times smaller error when compared to original calculations. When investigating the performance of MP calculations, it is determined that the time needed for all calculations to finish depends linearly on the number of modeled systems.

**Keywords:** dynamical systems, metabolism, biological computing.

## Introduction

The simplest natural living organism is a cell. It has evolved through natural ways for billions of years to do one simple operation – consume substances, transform them to new substances by chemical reactions and remove excess substances from the cell. This process is a basis for most biological organisms on Earth. If this process can be modeled accurately and efficiently, it can help solve challenging biological and medical problems.

The key element enabling and regulating substance exchange in a cell is a membrane. Computational model based on membrane structures was proposed in 2000 by Gheorghe Păun. He described a structure consisting of several membranes. Between these membranes are placed evolving objects. As it is described by G. Păun (2000), "each object can be transformed in other objects, can pass through a membrane, or can dissolve the membrane in which it is placed". By using several objects in a membrane and applying evolution rules to them, a computational model is obtained. This model is called a P system.

The new model quickly became a topic of research and many different specialized modifications to P system were developed. P systems with string objects processed by evolution rules distributed alongside the transitions of an Eilenberg machine are called EP systems (Bălănescu *et al*. 2003). System based on boundary rules are called PB systems (Bernardini, Manca 2003). A P system which uses classical evolution rules without communication targets is called Evolution-Communication P System (Cavaliere 2003). CSPS is a P system based on client and server membranes (Ciobanu *et al*. 2003a). Self-activating P systems are used for solving NP complete problems (Czeizler 2003). SN P systems incorporate spiking neurons to P system calculations (Ionescu *et al*. 2006).

P systems have many kinds of applications not limited to biology. These systems are so universal that they have been adapted for many purposes. For example, two algorithms were proposed to solve cryptographic problem with message authentication (Atanasiu 2003) and membrane systems were presented as a model for distributed computing (Ciobanu *et al*. 2003b). P systems were even proposed for linguistic applications (Enguix 2003). The new and interesting class of P system is Metabolic P system (MP). This system is based on metabolic process which is used by living organisms to consume matter, transform it and expel excess waste. MP systems are a special class of P systems introduced for expressing metabolism in a discrete mathematical setting (Manca 2010).

One problem is that there are no known implementations of MP system in dedicated hardware. Most implementation ways are limited to MP system simulation in

personal computers. Therefore, the possibility of such system implementation in hardware is unknown.

In this article we will review MP system development, including key publications, and discuss basic principles of MP system operation to select the most appropriate algorithms, models and tools for the experimental research. Implementation ways and real world application of MP systems will also be mentioned.

In the experimental section of this article, we will try to determine the possibility of implementing MP system model in a fixed point calculation capable hardware. This should help to determine, what hardware specifications are needed to effectively implement MP systems. We will also look in to the scalability of MP system calculations, because there can be a potential need to run many similar calculations at the same time.

**Development of MP system**

The first theoretical proposal of MP system was made by (Manca *et al*. 2005). In this paper it was proposed to observe rewriting rules in membranes from a different viewpoint than in P systems. It is done by defining a metabolic algorithm for computing the evolution of P systems when some initial state and some reaction parameters are given. These parameters can be, for example, reactivities and growing factors. As it is explained in the paper, the proposed metabolic algorithm "is inspired by a chemical reading of the rewriting rules":

$$\Delta\|X\| = \sum_{r\in R} \Delta_r \|X\|. \tag{1}$$

As defined in (Manca *et al*. 2005) Eq. 1 describes the transition from rewriting rules to metabolic equations. To compute the overall molar variation if an object $X$, contributions of all rules are taken into account by summing up their effects on the concentration of $X$, where $R$ is a set of rules in P system. An example of rule translation is also provided in the same paper. The following set of rules:

$$\begin{aligned} r1 &: AC \to AB; \\ r2 &: BC \to A; \\ r3 &: BBB \to BC, \end{aligned} \tag{2}$$

is associated to coefficients $k_{r1}$, $k_{r2}$ and $k_{r3}$, and can be transformed by applying (1) to a set of following metabolic equations:

$$\begin{aligned} \Delta\|A\| &= \ 0\cdot k_{r1}\|AC\| + 1\cdot k_{r2}\|BC\| + 0\cdot k_{r3}\|BBB\|; \\ \Delta\|B\| &= +1\cdot k_{r1}\|AC\| - 1\cdot k_{r2}\|BC\| - 2\cdot k_{r3}\|BBB\|; \\ \Delta\|C\| &= -1\cdot k_{r1}\|AC\| - 1\cdot k_{r2}\|BC\| + 1\cdot k_{r3}\|BBB\|. \end{aligned} \tag{3}$$

MP system with flux regulation maps is described by Manca (2009). It can be specified by a construct:

$$M = (X, R, V, Q, F, \nu, \mu, \tau, q_0, \delta), \tag{4}$$

where $X$ is the set of substances, $R$ is the set of reactions, $V$ is the set of parameters, $Q$ is the set of states, $F$ is the set of flux maps, $v$ is a natural number which specifies the number of molecules, μ is a function which assigns the mass, $\tau$ is the temporal interval, $q_0$ is the initial state and δ is the dynamics of the system. MP system without the set $Q$ of states and the dynamics is an MP graph (Manca 2009). When elements τ, $v$ and μ are also omitted, the resulting thing is called an MP grammar.

There are currently two major implementation ways of MP systems. The first one started as a *Psim* (Bianco *et al.* 2007). It is a simulation tool developed for MP system modeling. It allows to describe a system by means of MP graphs and simulate this systems dynamics based on metabolic algorithm. *Psim* is developed using Java programming language and features an input GUI which is used to draw appropriate elements and construct MP graphs. A finished graph can then be simulated and the results can be displayed in a chart form.

An improved software, called *MetaPlab*, was introduced in (Castellini, Manca 2009). It introduced a new plugin-based architecture which makes the software more versatile and able to perform multiple tasks. In the paper it is described as "virtual laboratory". *MetaPlab* can have many plugins and modifications to help perform the simulation as well as represent the data. There has also been a proposal to represent MP system data in XML format (Manca, Marchetti 2009). *MetaPlab* software is available under GPL open-source license and can be freely downloaded from *MetaPlab*.

Another implementation of MP systems is an open-source Java library appropriately called *MpTheory Java Library*. It is developed by V. Manca and L. Marchetti and is also available for download. The provided Java objects can be directly used to model selected MP systems, but the library can also be used within MATLAB, GNU Octave, Mathematica and R computing environments. This makes the library very versatile and compatible with many systems. As an output it produces a plot of the modeled MP system.

In this papers experimental section we will use our own implementation of selected MP system in MATLAB computing environment. We will do this to have greater control of the ways in which MP system calculations are performed. We need this to be able to simulate fixed

point environment. To do this in earlier described systems it would require substantial modifications to Java code and it would not be productive because we only intend to use one relatively simple MP system model.

### Applications of MP system

From the introduction of MP system, there have been many applications which demonstrate the usefulness of this system. One of the first biological processes modeled by MP systems was the Belousov-Zhabotinsky reaction (in the Brusselator formulation) (Bianco *et al*. 2006a). As described by Manca (2013), it has the following form:

$$\begin{aligned} a &\rightarrow x; \\ 2x+y &\rightarrow 3x; \\ b+x &\rightarrow y+d; \\ x &\rightarrow e. \end{aligned} \tag{5}$$

MP grammar of Brusselator reaction can be written by simplifying (5) reactions by focusing on substances $x$, $y$ and by considering $a$, $b$, $d$, $e$ as input/output substances (Table 1). There are, of course, many more biological processes which dynamics have been modeled by MP systems, including the Lotka-Volterra dynamics and the Susceptible-Infected-Recovered epidemic (Bianco *et al*. 2006b), the circadian rhythms, the mitotic cycles in early amphibian embryos (Manca, Bianco 2008), a Pseudomonas quorum sensing model (Bianco *et al*. 2006c), the *lac* operon gene regulatory mechanism in glycolytic pathway (Castellini *et al.* 2009).

Recently, Goldbeter's mitotic oscillator was modeled by MP system (Manca, Marchetti 2010). In this paper it was shown that metabolic P systems yield a robust method for biological modeling. In (Manca *et al*. 2011) MP systems were applied for developing new physiologically based models of the glucose-insulin system which can be applied to the Intra Venous Glucose Tolerance Test. New models based on MP system theory are being developed. For example, very recently MP systems were applied to breast cancer research (Bollig-Fischer *et al.* 2014). In the future we can expect more biological processes to be modeled by MP system theory.

Table 1. Brusselator MP grammar (source: Manca 2013)
1 lentelė. Briuseliatoriaus MP gramatika (šaltinis: Manca 2013)

| Reactions | Regulators |
|---|---|
| $r_1 : \varnothing \rightarrow x;$<br>$r_2 : 2x+y \rightarrow 3x;$<br>$r_3 : x \rightarrow y;$<br>$r_4 : x \rightarrow \varnothing.$ | $\varphi_1 = 1;$<br>$\varphi_2 = 10^{-6} x^2 y;$<br>$\varphi_3 = 0.03x;$<br>$\varphi_4 = 0.01x.$ |

In the experimental section we will focus on the Brusselator model. This model was selected because it is: one of the first processes modeled by MP systems, well established and understood. Also it is not very complicated when comparing to other MP system models. It was implemented as a standalone MATLAB function, but the modeling results can be applied to most MP systems.

### Experimental investigation

We will use MATLAB computing environment to determine the impact of using fixed point arithmetic to the accuracy of MP system modeling results. We will also try to determine the scalability of multiple simultaneous MP system simulations. These parameters are important to understand before attempting to implement MP systems in dedicated hardware platforms.

Our experimental setup consists of a personal computer with Intel i7-5820k hexa-core CPU (clocked at 3.8 GHz) and 16 GB DDR4 RAM using MATLAB R2014a software. In experiments, standard floating point double precision and fixed point calculations are used.

We have implemented the MP system and found that for fixed point calculations to work, at least 32 bit precision is needed. As shown in Figure 1, when using 16 bit precision, modeling results in constant value throughout the timeline. When experimenting with different bit counts for the fractional part, we found that the best results for this particular model are achieved when assigning 20 bits out of 32 for the fractional part. Therefore, throughout our experimentation we will be using 20 bits for the fractional part.

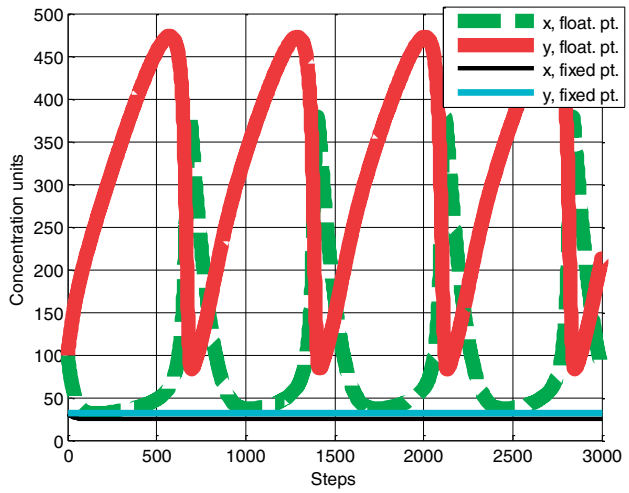


Fig. 1. Signals of the Brusselator modeled using floating point and 16 bit fixed point calculations

1 pav. Briuseliatoriaus signalai sumodeliuoti naudojant slankaus ir fiksuoto kablelio (16 bitų tikslumo) skaičiavimus

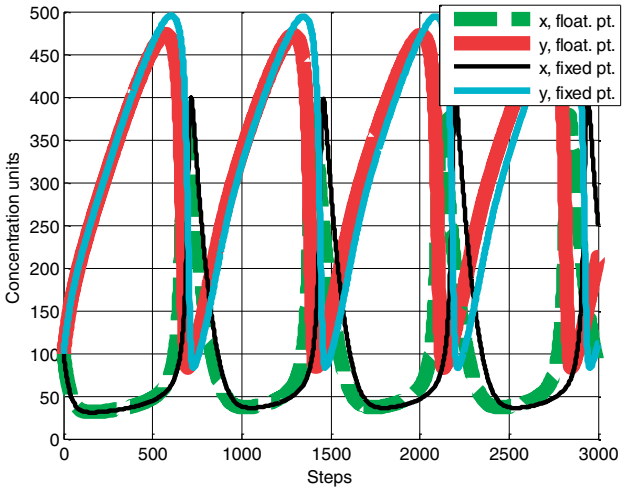


Fig. 2. Signals of Brusselator modeled using floating point and 32 bit fixed point calculations

2 pav. Briuseliatoriaus signalai sumodeliuoti naudojant slankaus ir fiksuoto kablelio (32 bitų tikslumo) skaičiavimus

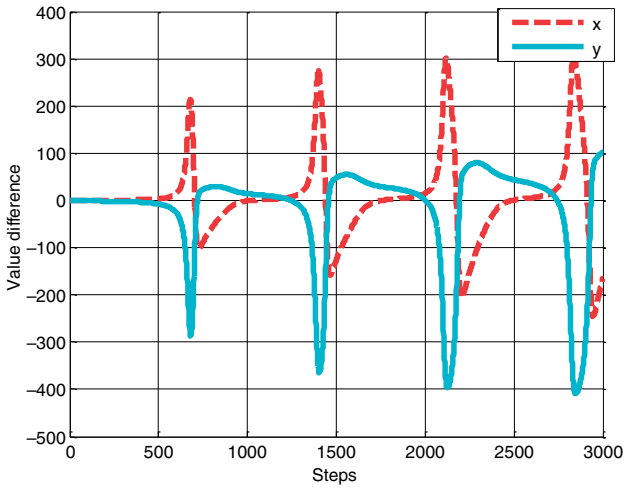


Fig. 3. $x$ and $y$ value difference when calculated using floating point and fixed point calculations

3 pav. $x$ ir $y$ reikšmių skirtumas, kai skaičiuojama naudojant slankų ir fiksuotą kablelį

Figure 2 shows the results when using 32 bit precision. In the first period of oscillation the difference between floating and fixed point is minimal, but it is increased in each following period. This results in the loss of precision, because each following point in time is:

$$x(t+1) = x(t) + \Delta t, \tag{6}$$

here new $x$ value always depends on the previous one.

The $\Delta t$ in (6) is a product of a stoichiometric matrix and a flux vector, calculated from MP grammar shown in Table 1. The full formula used for our MP system is

$$\begin{bmatrix} x(t+1) \\ y(t+1) \end{bmatrix} = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} 1 & 1 & -1 & -1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 10^{-6} x(t)^2 y(t) \\ 0.03x \\ 0.01y \end{bmatrix}. \tag{7}$$

MP systems, modeled by (7) in 32 bit fixed point calculations, precision is decreasing over time (Figure 3). At times the inaccuracy can reach more than 400 points – close to the total amplitude of the oscillation. It is caused by the specifics of the modeled system, as the oscillation includes some sharp changes of concentration values.

Table 2. Modified MP grammar of Brusselator reaction

2 lentelė. Modifikuota Briuseliatoriaus reakcijos MP gramatika

| Reactions | Regulators |
|---|---|
| $r_1 : \varnothing \to 0.01x;$ | $\varphi_1 = 100;$ |
| $r_2 : 0.02x + 0.01y \to 0.03x;$ | $\varphi_2 = 0.01 \cdot x \cdot x \cdot 0.01 \cdot y;$ |
| $r_3 : 0.01x \to 0.01y;$ | $\varphi_3 = 3x;$ |
| $r_4 : 0.01x \to \varnothing.$ | $\varphi_4 = x.$ |

To increase the accuracy of fixed point calculations, we have made an adjustment to Brusselator reactions MP grammar. The main purpose of this adjustment is to avoid having numbers with many digits, because they require more bits to be represented accurately in digital systems. The resulting MP grammar is shown in Table 2.

By applying modified MP grammar to our calculations, we get improved accuracy when using fixed point calculations. The modeling results, as shown in Figure 4, are very close between floating point and fixed point systems. Therefore, we can conclude that our modification of MP grammar is an effective way of improving fixed point calculation accuracy. The basis of our modification is a decreased amplitude of constants in MP grammar. This is achieved by dividing all constants in the reactions side of the table and multiplying the constants in the regulators side of the table. As long as the multiplier and divisor are the same, the resulting MP grammar is equivalent to the original. When using modified MP grammar the difference, shown in Figure 5, between floating and fixed point calculations is more than 250 times lower. This results in greatly increased accuracy and prevents the oscillation from amplifying the error.

The shown MP grammar modification method can be applied to most other MP systems. The main rule when adjusting the formula is to multiply regulators and divide reactions by the same amount. Of course, like in any other limited precision calculations, it is beneficial to avoid multiplying very small or very large numbers. In our modeling we avoided this when calculating $\varphi_2$ by adjusting the order of multiplied values.

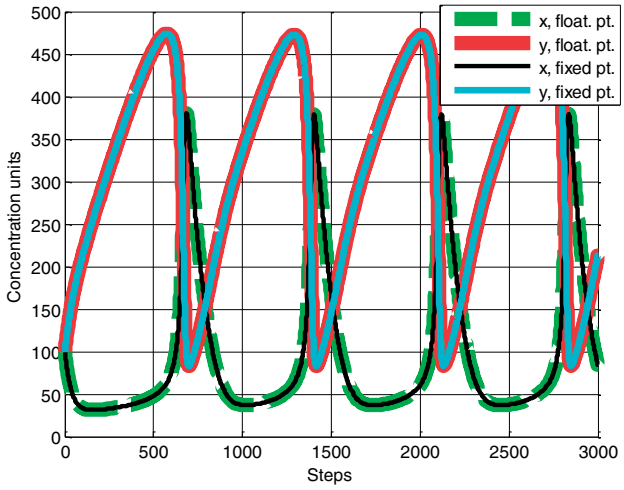


Fig. 4. Signals of the Brusselator modeled using floating point and 32 bit fixed point calculations using modified MP grammar

4 pav. Briuseliatoriaus signalai sumodeliuoti naudojant slankaus ir fiksuoto kablelio (32 bitų tikslumo) skaičiavimus, kai taikoma modifikuota MP gramatika

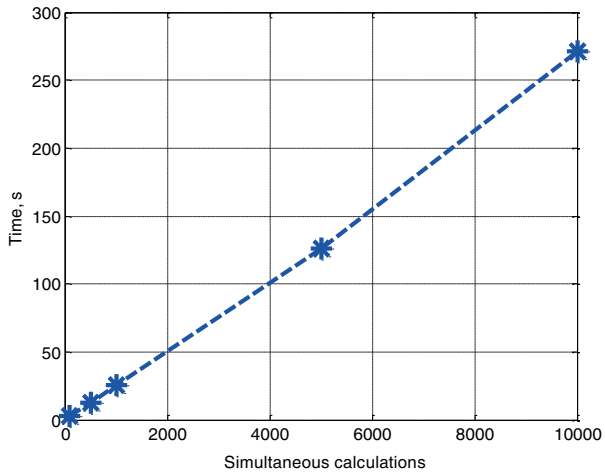


Fig. 6. Total calculation time depending on the number of simultaneously calculated models

6 pav. Skaičiavimo laiko priklausomybė nuo MP sistemų modelių skaičiaus

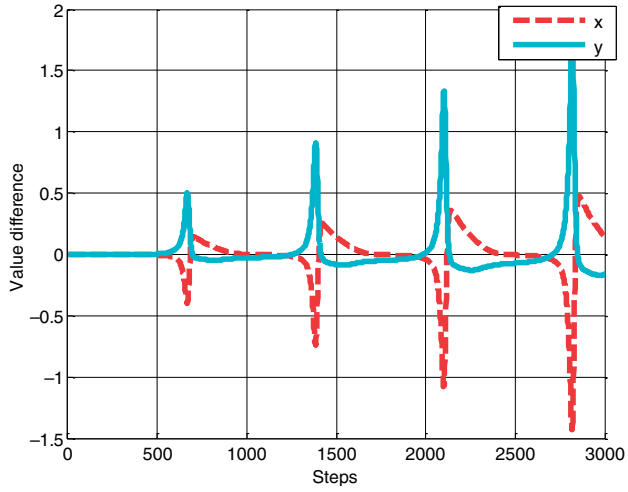


Fig. 5. Difference of x and y value calculated using floating point and fixed point calculations and using modified MP grammar

5 pav. Reikšmių x ir y skirtumas, kai skaičiuojama naudojant slankų ir fiksuotą kablelį bei modifikuotą MP gramatiką

Next, we evaluated the hardware performance when multiple MP system modeling calculations were executed at the same time. This scenario is often used when looking for suitable parameters of dynamical systems. Total calculation time increases linearly depending on how many systems are being modeled at the same time (Fig. 6). This result was expected because linear algebra calculations of each model are independent. The total time when simultaneously modeling 10,000 independent MP systems exceeds 250 s. In our case, we used the Brusselator model which is relatively simple, but when modeling more complex systems it is possible for the total time to increase greatly. In that case, parallel computing hardware should be considered.

## Conclusions

We have reviewed MP system development and implementation and selected an appropriate model and tools for our experimental investigation. After performing simulation of the MP system of Brusselator reaction we have concluded the following:

1. Bit count in fixed point arithmetic has a major impact on MP system modeling accuracy. When implementing MP systems in hardware, this need to be taken into account for each MP model.
2. Minimal recommended fixed arithmetic bit count for the Brusselator model is 32 bits. The best results are achieved when using 20 out of 32 bits for the fractional part.
3. Inaccuracies caused by using fixed point arithmetic are amplified on each oscillation period. The main reason is the specifics of linear algebra calculations in MP systems.
4. Proposed a rule to modify MP grammar for increased calculation accuracy: multiply regulators and divide reactions by the same amount to decrease the bit count required to accurately store MP grammar constants.
5. By applying modified MP grammar equation, the error of calculations is decreased more than 250 times. Fixed point calculation results become practically identical to floating point results.
6. Calculation time increases linearly depending on number of simultaneous calculations. Parallel computing should be considered for high number of simultaneous calculations.

## References


Atanasiu, A. 2003. Authentication of messages using P systems, *Lecture Notes in Computer Science* 2597: 33–42. http://dx.doi.org/10.1007/3-540-36490-0_3

Bălănescu, T.; Gheorghe, M.; Holcombe, M.; Ipate, F. 2003. Eilenberg P systems, *Membrane Computing*. Springer Berlin Heidelberg, 43–57. http://dx.doi.org/10.1007/3-540-36490-0_4

Bernardini, F.; Manca, V. 2003. P systems with boundary rules, *Membrane Computing*. Springer Berlin Heidelberg, 107–118. http://dx.doi.org/10.1007/3-540-36490-0_8

Bianco L.; Fontana, F.; Manca, V. 2006a. P Systems with reaction maps, *International Journal of Foundations of Computer Science* 17(1): 27–48. http://dx.doi.org/10.1142/S0129054106003681

Bianco L.; Fontana, F.; Franco, G.; Manca, V. 2006b. P Systems for biological dynamics, *Applications of Membrane Computing*. Springer Berlin Heidelberg, 83–128. http://dx.doi.org/10.1007/3-540-29937-8_3

Bianco L.; Pescini, D.; Siepmann, P.; Krasnogor, N.; Romero-Campero, F.; Marian, G. 2006c. Towards a P systems pseudomonas quorum sensing model, *Membrane Computing*. Springer Berlin Heidelberg, 197–214. http://dx.doi.org/10.1007/11963516_13

Bianco, L.; Manca, V.; Marchetti, L.; Petterlini, M. 2007. Psim: a simulator for biomolecular dynamics based on P Systems, in *IEEE Congress on Evolutionary Computation CEC 2007,* 25–28 September 2007, Singapore, 883–887. http://dx.doi.org/10.1109/CEC.2007.4424563

Bollig-Fischer, A.; Marchetti, L.; Mitrea, C.; Wu, J.; Kruger, A.; Manca, V.; Draghici, S. 2014. Modeling time-dependent transcription effects of HER2 oncogene and discovery of a role for E2F2 in breast cancer cell-matrix adhesion, *Bioinformatics* 30(21): 3036–3043. http://dx.doi.org/10.1093/bioinformatics/btu400

Castellini, A.; Franco, G.; Manca, V. 2009. Toward a representation of hybrid functional petri nets by MP systems, *Natural Computing*. Springer Japan, 28–37. http://dx.doi.org/10.1007/978-4-431-88981-6_3

Castellini, A.; Manca, V. 2009. MetaPlab: a computational framework for metabolic P systems, *Membrane Computing*. Springer Berlin Heidelberg, 157–168. http://dx.doi.org/10.1007/978-3-540-95885-7_12

Cavaliere, M. 2003. Evolution-communication P systems, *Membrane Computing*. Springer Berlin Heidelberg, 134–145. http://dx.doi.org/10.1007/3-540-36490-0_10

Ciobanu, G.; Dumitriu, D.; Huzum, D.; Moruz, G.; Tanas, B. 2003a. Client-server P systems in modeling molecular interactions, *Membrane Computing*. Springer Berlin Heidelberg, 203–218. http://dx.doi.org/10.1007/3-540-36490-0_13

Ciobanu, G.; Desai, R.; Kumar, A. 2003b. Membrane systems and distributed computing, *Membrane Computing*. Springer Berlin Heidelberg, 187–202. http://dx.doi.org/10.1007/3-540-36490-0_12

Czeizler, E. 2003. Self-activating P systems, *Membrane Computing*. Springer Berlin Heidelberg, 234–246. http://dx.doi.org/10.1007/3-540-36490-0_15

Enguix, G. B. 2003. Preliminaries about some possible applications of P systems in linguistics, *Lecture Notes in Computer Science* 2597: 74–89. http://dx.doi.org/10.1007/3-540-36490-0_6

Ionescu, M.; Păun, G.; Yokomori, T. 2006. Spiking neural P systems, *Fundamenta informaticae* 71(2): 279–308.

Manca, V. 2009. Fundamentals of metabolic P systems, *Handbook of membrane computing* 19. Oxford University Press, 475–498.

Manca, V. 2010. From P to MP systems, *Membrane Computing*. Springer Berlin Heidelberg, 74–94. http://dx.doi.org/10.1007/978-3-642-11467-0_7

Manca, V. 2013. *Infobiotics: information in biotic systems*. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-36223-1

Manca, V.; Bianco, L.; Fontana, F. 2005. Evolution and oscillation in P systems: applications to biological phenomena, *Lecture Notes in Computer Science* 3365: 63–84. http://dx.doi.org/10.1007/978-3-540-31837-8_4

Manca, V.; Marchetti, L.; Pagliarini, R 2011. MP modeling of glucose-insulin interactions in the intravenous glucose tolerance test, *International Journal of Natural Computing Research (IJNCR)* 2(3): 13–24. http://dx.doi.org/10.4018/jncr.2011070102

Manca, V.; Bianco, L. 2008. Biological networks in metabolic P systems, *BioSystems* 91(3): 489–498. http://dx.doi.org/10.1016/j.biosystems.2006.11.009

Manca, V.; Marchetti, L. 2009. XML representation of metabolic P systems, in *IEEE Congress on Evolutionary Computation CEC 2009,* 18–21 May 2009, Trondheim, Norway, 3103–3110. http://dx.doi.org/10.1109/CEC.2009.4983336

Manca, V.; Marchetti, L. 2010. Goldbeter's mitotic oscillator entirely modeled by MP systems, *Lecture Notes in Computer Science* 6501: 273–284. http://dx.doi.org/10.1007/978-3-642-18123-8_22

Păun, G. 2000. Computing with membranes, *Journal of Computer and System Sciences* 61(1): 108–143. http://dx.doi.org/10.1006/jcss.1999.1693


## METABOLINĖS P SISTEMOS TAIKYMO GALIMYBĖS


**D. Kulakovskis**



Santrauka

Metabolinė P sistema yra P sistemos klasė, pagrįsta metabolinio proceso, būdingo gyviesiems organizmams, taikymu. Siekiant pasirengti įgyvendinti MP sistemas aparatine įranga, pateikiama MP sistemų kūrimo, įgyvendinimo ir modeliavimo literatūros analitinė apžvalga. Įvertinama galimybė modeliuoti MP sistemas taikant fiksuotojo kablelio aritmetiką. Imitaciniais eksperimentais nustatytas minimalus pasirinktam MP sistemos modeliui taikyti reikiamas žodžių ilgis. Pasiūlytas būdas fiksuotosios aritmetikos skaičiavimo tikslumui padidinti. Šio būdo esmė yra MP gramatikos modifikavimas pagal konstantų daugybos ir dalybos taisyklę. Skaičiavimus atliekant modifikuota sistema, paklaida esti daugiau nei 250 kartų mažesnė. Tiriant MP sistemų našumą nustatyta, kad skaičiavimo trukmė tiesiškai priklauso nuo modeliuojamų sistemų skaičiaus.



**Reikšminiai žodžiai:** dinaminės sistemos, metabolizmas, biologinė kompiuterija.