INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(1):105-112, February 2016.

Membrane Computing and Economics: A General View

G. Păun

Gheorghe Păun
Institute of Mathematics of the Romanian Academy
PO Box 1-764, 014700 Bucureşti, Romania
gpaun@us.es

Abstract: Three are the points we briefly discuss here: using membrane computing tools for efficient computing/optimization, the possibilities of using “general" membrane computing (P systems using multisets of symbol objects processed by biochemical-like evolution rules) as a framework for modeling economic processes, and the numerical P systems, a class of computing devices explicitly defined with a motivation related to economics. The discussion is rather informal, only pointing out research directions and providing bibliographical information.

Keywords: membrane computing, P system, economics, numerical P system.

1 A Quick Glimpse to Membrane Computing

Membrane computing is a branch of natural computing aiming to abstract computing models from the structure and the functioning of the living cells – at least this was the initial motivation of the research in this area, [12]. Basically, in the compartments of a cell-like membrane structure one places multisets (sets with multiplicities associated with their elements) of *objects*, represented by symbols of a given alphabet, and *evolution rules*, in the form of “reactions" transforming certain objects in other objects (multiset rewriting rules, written in the form $u \to v$, where u and v are multisets); the objects can also pass through membranes, from a compartment to another one. Starting from an initial configuration (of multisets and membranes) and applying the evolution rules according to specified strategies (semantics), one obtains transitions among configurations, hence *computations.* In this way we get a computing device, usually called a *P system.* Results are associated with computations, in various ways.

Most investigated in membrane computing are P systems with the rules applied in the non-deterministically maximally parallel way (a multiset of rules is non-deterministically chosen and applied in each multiset, such that there is no applicable multiset which contains strictly the chosen multiset) and the result associated only with halting computations, those which reach a configuration where no evolution rule can be applied.

There are many variations of this basic model. First, one can consider various forms of the rules, as well as ways of controlling their application: priority, promoters, inhibitors, periodical change of rules in time, etc. Then, other ways of using the rules can be considered: sequential, limited parallelism, minimal parallelism, asynchronous systems, etc. Also the result of a computation can be defined in various ways (internally, externally, in the form of a number or of a string), not necessarily for halting computations.

Further suggestions come from biology. For instance, instead of multiset rewriting rules, as those corresponding to biochemical reactions, we can consider symport or antiport rules, which move multisets of objects across membranes, [11]. Similarly, we can consider rules for handling also the membranes, not only the objects: membranes can be created, dissolved, divided, the relationships between membranes can change in timne (e.g., by exocytosis, phagocytosis, etc.), and so on.

Then, an extension can be considered from hierarchical arrangements of membranes, like in a cell, described by trees, to a tissue-like arrangement of membranes, described by a general graph. All ingredients mentioned above can be extended to this case.

In short, a framework for processing multisets (of symbol objects) in a distributed parallel way.

Instead of symbol objects, we can also consider more complex objects, strings for instance – and then the evolution rules should be chosen suitably.

Most of the classes of the P systems suggested above are equivalent as computing power with Turing machines, and this happens even for rather restricted (as the number of membranes or the complexity of the evolution rules) systems.

Furthermore, due to their massive parallelism, and to the possibility of creating an exponential workspace in a linear time, certain classes of P systems can solve computationally hard problems (typically, **NP**-complete problems) in a feasible (polynomial) time. (This has been called in [14] *fypercomputing*, as an extension from hypercomputing, with "f" coming from "fast".) This is based on a time-space trade-off, with the space created by operations like membrane creation, membrane division, string duplication. Depending on the used ingredients, even (polynomial) characterizations of **PSPACE** are obtained.

Two classes of P systems different from the previous ones are the *numerical P systems*, which will be discussed below, and the *spiking neural P systems* (in short, SN P systems), [7], inspired from the way the neurons cooperate in a brain, communicating through spikes, electrical impulses of identical shape. The results are the same also for SN P systems: computational universality and the possibility of *fypercomputing* in the case when neuron division or similar operations are present. Another interesting idea leading to fypercomputations in this area is to use (arbitrarily large) pre-computed resources, with a limited amount of information present in the initial configuration.

Besides these theoretical directions of investigations, concerning the power and the efficiency of P systems, well developed is the application area, naturally, starting with applications in biology and biomedicine. Many other applications are already reported, in ecology, linguistics, computer graphics, cryptography, approximate optimization, robot control – as well as to economics.

The literature of membrane computing is rather large. Details can be found at the domain web site from `http://ppage.psystems.eu`. An introduction can be found in [13], a comprehensive coverage (at the level of 2009) in [18], while applications can be found in [4], [6], [20]. Many collective volumes, PhD theses and downloadable papers can be found at the above mentioned web address.

2 Using Membrane Computing Tools in Economics

Two ideas mentioned in the previous section suggest already that membrane computing can be useful for economics, especially in the computability, the operation research, and the optimization. The P systems as models will be discussed below, here we have in mind P systems as tools for handling "classic" models.

First, we mentioned the many results where **NP**-complete problems are solved in polynomial time by certain classes of P systems. Most of the non-trivial practical problems appearing in economics (as well as in other areas) are **NP**-complete. Thus, it would be great to solve them in a polynomial time by using a "cellular computer". However, the solutions provided by membrane computing are still *in info*, theoretical, as there is no implementation of P systems *in vitro/vivo*, not even *in silico* (such that the massive parallelism can be effectively used). There are implementations of P systems on parallel hardware, but not at the level of making practical the theoretical fypercomputing. On the other hand, it is highly probably that even a bio-implementation of a P system will not help too much, it will be similar to the case of DNA computing, with the 1994 Adleman experiment, which remained at the level of a *demo*, of solving toy problems, without practical consequences.

Much more practical is the possibility of using the so-called *membrane algorithms* in solving computationally hard optimization problems. This is a rather developed direction of research in membrane computing, with a large bibliography and very encouraging results. Basically, this is nothing else than distributed evolutionary computing, but with the distribution controlled by means of membranes, and with further ingredients from membrane computing used, such as membrane creation, division, dissolution, communication among membranes, neural-like organization of cells, etc. Many engineering applications of membrane algorithms were reported – the reader is advised to consult [4], [6], [18], the forthcoming volume [20], as well as the bibliography from the membrane computing web site.

3 P Systems as Models for Economics

We pass now to a more particular approach, namely to consider membrane computing as a framework for building models for economic processes.

Of course, the first question which could be formulated is: why? There are so many (mathematical) frameworks already used in economics, why considering a new, somewhat exotic (for instance, by its biological inspiration), one? The answer is complex. In general, new tools are always good to be checked, maybe they will bring new possibilities to address old problems in a new framework, ideally, to also formulate new questions in the new framework. Membrane computing is not only *natural* in a direct sense, but also sufficiently developed at the theoretical, abstract level that it is just expected to be applicable in many areas, apparently far away from biology. Then, the biological metaphor is very general and very useful, as indicated by the many applications of membrane computing in areas rather different from one another.

More specifically, there are several features of P systems which make them attractive as models, for economics like for biology, too. They pertain to discrete mathematics, so that they are adequate to situations when we have to deal with small numbers of objects and agents. Applying tools of continuous mathematics (especially differential equations) in situations which are clearly of a discrete type leads to wrong results. Then, P systems are easily scalable, easily understandable, easily programmable – three appealing properties. The behavior of a (non-trivial) P system is “emergent", impossible to be predicted by simply examining the rules. (The universality also implies, through Rice theorem, that no non-trivial question about a P system can be answered algorithmically, hence a computer simulation is needed.) Finally, there are many economic issues which can be easily and transparently captured in terms of membrane computing. Membranes can describe the organization, the multiset rewriting rules and the symport/antiport rules can describe production and commercial operations.

Only a hint, recalling some lines from [16].

Assume that units of row material a are introduced on a market by a supplier S and are used by producers P_i, each having a number of production units b_i, to manufacture goods d_i; these goods are then sold by P_i at price p_i to retailers R_j, each having a number of capacity units c_j; in turn, the retailers sell the goods to a general consumer C, which introduces the need for d, denoted by $\bar{d}$, on the market. For instance, by rules

$$S \to Sa^n, \quad C \to C\bar{d}^m,$$

the supplier introduces n units of a and the consumer introduces the need for m copies of d.

Producer P_i can produce one d by consuming one a and one b_i,

$$b_i a \to d_i,$$

while retailer R_j takes one order for d by means of

$$c_j \bar{d} \to \bar{d}_j.$$

(The good d and the need for d are indexed with the label of the producer – who fix its price – and of the retailer, respectively.) Then, the producers and the retailers interact: by

$$d_i \bar{d}_j \to b_i c_j \ \ [p_{i,j}],$$

one copy of good d passes from P_i to R_j, because R_j has an order for this good, and this happens with the probability $p_{i,j}$, which depends on the previous transactions between producer P_i and retailer R_j; units of capacities, b_i and c_j, are free again.

We have not yet included money here, but this can be easily done. Instead of $b_i a \to d_i$, we have to consider the rule

$$b_i a u_i^r \to d_i u_S^r,$$

with the meaning that P_i has paid r monetary units (denoted by u_i when belonging to P_i) for a, thus S has gained r copies of u_S (notation for a monetary unit in possession of S). In turn, instead of $d_i \bar{d}_j \to b_i c_j \ \ [p_{i,j}]$, we have to use

$$d_i \bar{d}_j v_j^{p_i} \to b_i c_j u_i^{p_i} \ \ [p_{i,j}],$$

with the meaning that p_i monetary units have passed from the "account" (= multiset) of R_j to that of P_i. The retailer will then sell the good at a price (if possible) greater than p_i to the general consumer C, and the cycle is repeated.

The model can still be enlarged: for instance, the number of a's and $\bar{d}$'s introduced on the market can fluctuate, the prices producers and retailers set can vary in time; then, both producers and retailers can make investments (transform money into production/storing units), maybe in a prudent manner (with low probability associated with investment rules and higher probability associated with saving rules, of the form $u_i \to u_i$, just keeping the money unused), with or without limits on the total number of investments, and so on.

It is important to note that coefficients $p_{i,j}$ written in square brackets on the right hand side of the rules are not "pure" probabilities, but they also express the trust of retailers when buying from producers, depending on the history of their collaboration. These coefficients can be changed over time, which adds to the complexity of the model.

The reader is refereed to [15], [16], [17] for further details, illustrations, and discussions.

Several other titles are provided in the bibliography, where economic processes are approached in terms of P systems.

After writing such a model, it is necessary to simulate it on a computer, hence a program is needed (ideally, implemented on a parallel support). Of course, the model is not used as a computing one, but the evolution itself of the model is of interest (this is similar to the applications of P systems in biology).

We are convinced that this direction of research deserves further efforts.

4 Numerical P Systems

This class of P systems was introduced (in paper 16 from the list in the next section) with an explicit economic inspiration. Briefly speaking, in the regions of a cell-like membrane structure, we have *numerical variables* (not "chemicals", as in the P systems discussed above), evolving by means of *production-repartition programs*. The variables from region i are denoted by

$x_{1,i}, x_{2,i}, \ldots, x_{k_i,i}$. If the model is used as a computing devices, the variables take integer values, but for applications the values can be real numbers.

A production-repartition program (from region i) is of the form

$$F_i(x_{1,i}, \ldots, x_{k_i,i}) \longrightarrow c_1|v_1 + c_2|v_2 + \cdots + c_{n_i}|v_{n_i},$$

where $F_i(x_{1,i}, \ldots, x_{k_i,i})$ is the "production function" and $c_1|v_1 + c_2|v_2 + \cdots + c_{n_i}|v_{n_i}$ defines the "repartition protocol". Denote $C_i = \sum_{s=1}^{n_i} c_s$ and denote by $x_{i,j}(t)$ the value of variable $x_{i,j}$ at step $t \geq 0$ ($t = 0$ corresponds to the initial values of variables).

At any time $t \geq 0$, we compute $F_i(x_{1,i}(t), \ldots, x_{k_i,i}(t))$. The value $q = F_i(x_{1,i}(t), \ldots, x_{k_i,i}(t))/C_i$ represents the "unitary portion" to be distributed to variables $v_1, \ldots, v_{n_i}$, according to coefficients $c_1, \ldots, c_{n_i}$ in order to obtain the values of these variables at time $t+1$. Specifically, v_s will receive $q \cdot c_s, 1 \leq s \leq n_i$. If a variable receives such "contributions" from several neighboring compartments, then they are added in order to produce the next value of the variable. A variable used in a production function is consumed, reset to 0; if a variable does not appear in a production function, then its value remains unchanged – of course, it can change by receiving "production portions" (note that they can also be negative) from its region or from the neighboring regions. In this way, we pass from given values of the variables to next values. The process can be iterated. In this way, we get a computation. The positive values of a specified variable form the set of numbers *generated* by a numerical P system.

Again, further ingredients can be added – for instance, a control on the use of a program, such as the "enzymatic control" considered in several papers dealing with robot controllers based on numerical P systems, or restrictions can be considered – particular functions as production functions. As expected, these systems are both powerful (Turing complete) and efficient.

In the robot control area, numerical P systems were used as devices for computing functions of several real variables, and this seems to be the most plausible idea also for economic applications. Anyway, again we expect applications in the two directions mentioned also for usual P systems: as tools for efficient computing and as models.

Both at the theoretical level and in what concerns the applications, numerical P systems are waiting for further attention, and we believe that the effort will be rewarding. In order to help the reader in this direction, we provide below a list of all titles we know at this moment related to numerical P systems (but further research are in development).

5 A Bibliography of Numerical P Systems

1. O. Arsene, C. Buiu, N. Popescu: SNUPS, A simulator for numerical membrane computing. *Intern. J. of Innovative Computing, Information and Control*, 7, 6 (2011), 3509–3522.

2. C. Buiu, O. Arsene, C. Cipu, M. Pătraşcu: A software tool for modeling and simulation of numerical P systems. *Biosystems*, 103, 3 (2011), 442-447.

3. C. Buiu, C.I. Vasile, O. Arsene: Development of membrane controllers for mobile robots. *Information Sciences*, 187 (2012), 22–51.

4. M. Garcia-Quismondo, L.F. Macias-Ramos, M.J. Pérez-Jiménez: Implementing enzymatic numerical P systems for AI applications by means of graphic processing units. *Topics in Intelligent Engineering and Informatics*, vol. IV, Springer, Berlin, 2013, 137–159.

5. A. Leporati, A.E. Porreca, C. Zandron, G. Mauri: Improved universality results for parallel enzymatic numerical P systems. *Intern. J. Unconventional Computing*, 9 (2013), 384–404.

6. A. Leporati, G. Mauri, A.E. Porreca, C. Zandron: Enzymatic numerical P systems using elementary arithmetic operations. *Membrane Computing. Proc. 14th Intern. Conf., CMC 2013, Chişinău, August 2013*, LNCS 8340, Springer, Berlin, 2014, 249–264.

7. D. Llorente–Rivera, M.A. Gutiérrez–Naranjo: The pole balancing problem with enzymatic numerical P systems. *Proc. BWMC 2015*, Sevilla, February 2015, Fenix Editora, Sevilla, 2015 (in press).

8. A.B. Pavel: *Membrane Controllers for Cognitive Robots.* Master's thesis, Department of Automatic Control and System Engineering, Politehnica University of Bucharest, Romania, February 2011.

9. A.B. Pavel: *Development of Robot Controllers Using Membrane Computing.* PhD Thesis, Department of Automatic Control and System Engineering, Politehnica University of Bucharest, Romania, July 2015.

10. A.B. Pavel, O. Arsene, C. Buiu: Enzymatic numerical P systems – a new class of membrane computing systems. *The IEEE Fifth Intern. Conf. on Bio-Inspired Computing. Theory and applications. BIC-TA 2010*, Liverpool, Sept. 2010, 1331–1336.

11. A.B. Pavel, C. Buiu: Using enzymatic numerical P systems for modeling mobile robot controllers. *Natural Computing*, 11, 3 (2012), 387–393.

12. A.B. Pavel, I. Dumitrache: Hybrid numerical P systems. *UPB Scientific Bulletin*, Series A, submitted 2014.

13. A.B. Pavel, C.I. Vasile, I. Dumitrache: Robot localization implemented with enzymatic numerical P systems. *Proc. Conf. Living Machines 2012*, LNCS 7375, Springer, Berlin, 2012, 204–215.

14. Gh. Păun: Some open problems about numerical P systems. *Proc. 11th Brainstorming Week on Membrane Computing*, Sevilla, 4-8 February 2013, Fénix Editora, Sevilla, 2013, 235–242.

15. Gh. Păun: Some open problems about catalytic, numerical and spiking neural P systems, *Membrane Computing. Proc. 14th Intern. Conf., CMC 2013, Chişinău, August 2013*, LNCS 8340, Springer, Berlin, 2014, 33–39.

16. Gh. Păun, R. Păun: Membrane computing and economics: Numerical P systems. *Fundamenta Informaticae*, 73 (2006), 213–227.

17. Gh. Păun, R. Păun: Membrane computing as a framework for modeling economic processes. *Seventh Intern. Symp. Symbolic and Numeric Algorithms for Scientfic Computing*, SYNASC 2005, IEEE Press, 2005, 11–18.

18. Gh. Păun, R. Păun: Membrane computing and economics. Section 23.6 in *Handbook of Membrane Computing*, OUP, 2010, 632–644.

19. C.I. Vasile: *Distributed Control for Multi-Robot Systems.* PhD Thesis, Department of Automatic Control and System Engineering, Politehnica University of Bucharest, Romania, July 2015.

20. C.I. Vasile, A.B. Pavel, I. Dumitrache: Universality of enzymatic numerical P systems. *Intern. J. Computer Math.*, 90, 4 (2013), 869–879.

21. C.I. Vasile, A.B. Pavel, I. Dumitrache, J. Kelemen: Implementing obstacle avoidance and follower behaviors on Koala robots using numerical P systems. *Tenth Brainstorming Week on Membrane Computing*, Sevilla, 2012, vol. II, 215–227.

22. C.I. Vasile, A.B. Pavel, I. Dumitrache, Gh. Păun: On the power of enzymatic numerical P systems. *Acta Informatica*, 49, 6 (2012), 395–412.

23. X. Wang, G. Zhang, F. Neri, T. Jiang, J. Zhao, M. Gheorghe, F. Ipate, R. Lefticaru: Design and implementation of membrane controllers for trajectory tracking of nonholonomic wheeled mobile robots. *Integrated Computer-Sided Engineering*, to appear.

6 Concluding Remarks

In many circumstances, computer science got very fruitful inspiration from biology, and this became a systematic research direction in the last decades under the mane of *natural computing*. A comprehensive overview can be found in [19].

Membrane computing is part of this endeavor, well developed at the mathematical level, general, versatile, promising at the level of applications in very (apparently) different areas.

Here, we only mentioned some of the possibilities and the researches reported so far concerning the applications of membrane computing in economics, calling the attention to this research area, with the conviction that it deserves to be explored.

Bibliography

[1] J. Bartosik (2004); Paun's systems in modeling of human resource management. *Second Conference on Tools and Methods of Data Transformation*, WSU, Kielce, Poland..

[2] J. Bartosik (2004); Heaps of pieces and Paun's systems. *Second Conference on Tools and Methods of Data Transformation*, WSU Kielce.

[3] J. Bartosik, W. Korczynski (2002); Systemy membranowe jako modele hierarchicznych struktur zarzadzania, *Mat. Pokonferencyjne Ekonomia, Informatyka, Zarzadzanie. Teoria i Praktyka*, Wydzial Zarzadzania AGH, Tom II, AGH 2002.

[4] G. Ciobanu, Gh. Păun, M.J. Pérez-Jiménez (eds.) (2006); *Applications of Membrane Computing.* Springer, Berlin.

[5] R. Freund, M. Oswald, T. Schirk (2007); How a membrane agent buys goods in a membrane store. *Progress in Natural Science*, 17: 442–448.

[6] P. Frisco, M. Gheorghe, M.J. Pérez-Jiménez (eds.) (2014); *Applications of Membrane Computing in Systems and Synthetic Biology*, Springer, Berlin.

[7] M. Ionescu, Gh. Păun, T. Yokomori (2006); Spiking neural P systems. *Fundamenta Informaticae*, 71(2-3): 279–308.

[8] W. Korczynski (2004); On a model of economic systems. *Second Conference on Tools and Methods of Data Transformation*, WSU Kielce.

[9] W. Korczynski (2005); Păun's systems and accounting. *Pre-Proceedings of Sixth Workshop on Membrane Computing WMC6*, Vienna, Austria, 461–464.

[10] M. Oswald (2007); Independent agents in a globalized world modelled by tissue P systems. *Artificial Life and Robotics*, 11: 171–174.

[11] A. Păun, Gh. Păun (2002); The power of communication: P systems with symport/antiport. *New Generation Computing*, 20(3): 295–306.

[12] Gh. Păun (2000); Computing with membranes. *J. Computer and System Sciences*, 61(1): 108–143, first circulated as *TUCS Report 208*, November 1998 (www.tucs.fi).

[13] Gh. Păun (2002); *Membrane Computing. An Introduction.* Springer, Berlin.

[14] Gh. Păun (2012); Towards "fypercomputations" (in membrane computing). *Languages Alive. Essays Dedicated to Jurgen Dassow on the Occasion of His 65 Birthday* (H. Bordihn, M. Kutrib, B. Truthe, eds.), LNCS 7300, Springer, Berlin, 207–221.

[15] Gh. Păun, R. Păun (2005); Membrane computing as a framework for modeling economic processes. *Proc. SYNASC 05*, IEEE Press, 11–18.

[16] Gh. Păun, R. Păun (2006); A membrane computing approach to economic modeling: The producer-retailer interactions and investments. *Analiză şi Prospectivă Economică*, Part I: 3(1): 30–37, Part II: 4(2): 47–54.

[17] Gh. Păun, R. Păun (2006); Membrane computing models for economics. An invitation-survey. *Studii şi Cercetări de Calcul Economic şi Cibernetică Economică*, 40(1-2): 5–19.

[18] Gh. Păun, G. Rozenberg, A. Salomaa, eds.(2010); *Handbook of Membrane Computing.* Oxford University Press.

[19] G. Rozenberg, T. Bäck, J.N. Kok, eds. (2010); *Handbook of Natural Computing.* 4 vols., Springer, Berlin, 2012.

[20] G. Zhang, M. Gheorghe, M.J. Pérez-Jiménez (2016); *Real-life Applications with Membrane Computing.* Springer, Berlin (in press).