The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

Membrane Computing and Brain Modelling

Marion OSWALD
Faculty of Informatics, Vienna University of Technology, Vienna, Austria
marion@logic.at

Abstract

We briefly discuss variants of (extended) spiking neural P systems that combine features from the area of membrane computing and spiking neurons.

1 Introduction

P systems (membrane systems) were introduced by Gheorghe Păun in 1998 (see [16]) as a formal model describing the hierarchical structure of membranes in living organisms and the biological processes in and between cells (an introduction to this field can be found in the monograph [17], for the actual state of research we refer to the well maintained bibliography at [22]).

In the area of membrane computing there are two main classes of systems: P systems with a hierarchical (tree-like) structure as introduced in the seeding paper of this research area ([16]) and tissue P systems where the cells are arranged in an arbitrary graph structure (see [15]).

Based on the observation that neurons send electrical impulses (also called *spikes*) along axons to other neurons, new models in the area of neural computation were introduced, e.g., see [8], [13], [14].

Combining the ideas of (tissue) P systems and spiking neurons, spiking neural P systems (SNP systems for short) were introduced in [10]. For the motivation and the biological background of spiking neural P systems we refer the reader to [10], [18], here we just briefly recall that in SNP systems, the contents of a cell (neuron) consists of a number of so-called spikes; the rules assigned to a cell allow for sending information to other neurons in the form of electrical impulses (also called spikes) which are summed up at the target cell; the application of the rules depends on the contents of the neuron (which, in the general case, is described by regular sets). The system is synchronized but it works sequentially at the level of the neurons: in every step at most one rule is used in each of the neurons. As inspired by biological findings, the cell sending out spikes may be "closed" for a specific time period corresponding to the refraction period of a neuron; during this refraction period, the neuron is closed for new input and cannot get excited ("fire") for spiking again.

The length of the axon may cause a time delay before a spike arrives at the target. Moreover, the spikes coming along different axons may cause effects of different magnitude. These features have been included in the extended model of spiking neural P systems considered below.

This paper is intended to give a brief and informal overview of some of the models already proposed in the literature and is organized as follows.

In the next section, we give the original definition of ESNP systems as introduced in [1], before discussing some extended versions that incorporate other biologically inspired features. In section 4, the computational power of the proposed systems is summarized, while some further remarks in section 5 conclude this paper.

2 Definitions

$\mathbb{N}$ denotes the set of non-negative integers. The interval of non-negative integers between k and m is denoted by $[k..m]$.

In the following, we give the original definition of extended spiking neural P systems as they have been introduced in [1]:

An *extended spiking neural P system* (*ESNP system* for short) is a construct

$$\Pi = (m, S, R)$$

where

- m is the number of *cells* (or *neurons*); the neurons are uniquely identified by a number between 1 and m (obviously, we could instead use an alphabet with m symbols to identify the neurons);
- S describes the *initial configuration* by assigning an initial value (of spikes) to each neuron; for the

sake of simplicity, we assume that at the beginning of a computation we have no pending packages along the axons between the neurons;

- R is a finite set of *rules* of the form $\left(i, E/a^k \to P; d\right)$ such that $i \in [1..m]$ (specifying that this rule is assigned to cell i), $E \subseteq REG(\mathbb{N})$ is the *checking set* (the current number of spikes in the neuron has to be from E if this rule shall be executed), $k \in \mathbb{N}$ is the "number of spikes" (the energy) consumed by this rule, d is the *delay* (the "refraction time" when neuron i performs this rule), and P is a (possibly empty) set of *productions* of the form (l, w, t) where $l \in [1..m]$ (thus specifying the target cell), $w \in \mathbb{N}$ is the *weight* of the energy sent along the axon from neuron i to neuron l, and t is the time needed before the information sent from neuron i arrives at neuron l (i.e., the *delay along the axon*).

A *configuration* of the ESNP system is described as follows:

- for each neuron, the actual number of spikes in the neuron is specified;
- in each neuron i, there may be an "activated rule" $\left(i, E/a^k \to P; d'\right)$ waiting to be executed where d' is the remaining time until the neuron spikes;
- in each axon to a neuron l, we may find pending packages of the form (l, w, t') where t' is the remaining time until w spikes have to be added to neuron l provided it is not closed for input at the time this package arrives.

A *transition* from one configuration to another one now works as follows:

- for each neuron i, we first check whether we find an "activated rule" $\left(i, E/a^k \to P; d'\right)$ waiting to be executed; if $d' = 0$, then neuron i "spikes", i.e., for every production (l, w, t) occurring in the sequence r we put the corresponding package (l, w, t) on the axon from neuron i to neuron l, and after that, we eliminate this "activated rule" $\left(i, E/a^k \to P; 0\right)$;
- for each neuron l, we now consider all packages (l, w, t') on axons leading to neuron l; provided the neuron is not closed, i.e., if it does not carry an activated rule $\left(i, E/a^k \to P; d'\right)$ with $d' > 0$, we then sum up all weights w in such packages where $t' = 0$ and add this sum to the corresponding number of spikes in neuron l; in any case, the packages with $t' = 0$ are eliminated from the axons, whereas for all packages with $t' > 0$, we decrement t' by one;
- for each neuron i, we now again check whether we find an "activated rule" $\left(i, E/a^k \to P; d'\right)$ (with $d' > 0$) or not; if we have not found an "activated rule", we now may apply any rule $(i, E, k, d; r)$ from R for which the current number of spikes in the neuron is in E and then put a copy of this rule as "activated rule" for this neuron into the description of the current configuration; on the other hand, if there still has been an "activated rule" $\left(i, E/a^k \to P; d'\right)$ in the neuron with $d' > 0$, then we replace d' by $d' - 1$ and keep $\left(i, E/a^k \to P; d' - 1\right)$ as the "activated rule" in neuron i in the description of the configuration for the next step of the computation.

After having executed all the substeps described above in the correct sequence, we obtain the description of the new configuration. A *computation* is a sequence of configurations starting with the initial configuration given by S. A computation is called *successful* if it halts, i.e., if no pending package can be found along any axon, no neuron contains an activated rule, and for no neuron, a rule can be activated.

In the original model introduced in [10], in the productions (l, w, t) of a rule $\left(i, E/a^k \to \{(l, w, t)\}; d\right)$, only $w = 1$ (for *spiking rules*) or $w = 0$ (for *forgetting rules*) as well as $t = 0$ was allowed (and for forgetting rules, the checking set E had to be finite and disjoint from all other sets E in rules assigned to neuron i). Moreover, reflexive axons, i.e., leading from neuron i to neuron i, were not allowed, hence, for (l, w, t) being a production in a rule $\left(i, E/a^k \to P; d\right)$ for neuron i, $l \neq i$ was required. Yet the most important extension is that different rules for neuron i may affect different axons leaving from it whereas in the original model the structure of the axons (called synapses there) was fixed.

Depending on the purpose the ESNP system shall be used, some more features have to be specified: e.g., for generating k-dimensional vectors of non-negative integers, k neurons have to be designated as *output neurons*. There are several possibilities to define how the output values are computed; according to [10], the distance between the first two spikes in an output neuron defines its value; another possibility is to take the number of spikes at the end of a successful computation in the neuron as the output value (e.g., see [1]).

3 Some Natural Extensions

A quite natural feature found in biology and also used in the area of neural computation is that of inhibitory neurons or connections between neurons. The model of extended spiking neural P systems was extended again in [7] by considering inhibitory axons that allow for "closing" a neuron for one step by sending a spike along such an inhibitory axon to this neuron from another one. These systems were used to model logical gates in [7], but they also promise to be interesting for specifying other models of computation, e.g., the relation between extended spiking neural P systems with inhibitory axons and Petri nets could be interesting to explore.

Another model of membrane systems incorporating some specific features of complex systems consisting of two interacting networks of neurons and astrocytes has been investigated in [3]:

Astrocytes, a sub-type of macroglia have been understood as star-shaped glial cells spanning around neurons in the central nervous system (e.g., see [20]). Newer findings, however, have shown that a complex feedback loop of neuronal modulation exerted by astrocytes can be postulated. The influence of astrocytes in the functioning of the human brain has also been investigated in [21], where the influence of the capillary system in connection with the networks of neurons and astrocytes was modelled.

Inspired by this biological background (but without taking into account the capillary system), the concept of astrocytes influencing the signals along the axons has been incorporated in ESNP systems in [3] and [2]. For the astrocytes themselves, their membrane potential was assumed to be changed according to external inputs which may either come from neural cells ([3]) or from the firing intensity and frequency along the axon ([2]).

These models could well be used for the representation of artificial neural networks, especially for self-organizing feature maps; yet in contrast to analytic models of such variants of neural networks, the proposed models work in a discrete manner, but on the other hand, are based on a graph-like structure and not on a (usually two-dimensional) grid. (An example of such a two-dimensional artificial neural network based on biological observations of the complex networks of neurons and astrocytes in the human neocortex can be found in [4].)

For specific applications, especially in the area of artificial neural networks and self-organizing feature maps, an extended version where the dynamic evolution of new connections between neurons is allowed, could be useful; the influence of the already existing astrocytes on these new axons plays an important role. Another variant to be considered in the future are networks where part of the network may be destroyed which also has an interesting biological background. In this case, the ability of such a complex network to reorganize itself is the most challenging aspect of this variant.

4 Computational Power

(E)SNP systems are very powerful from a computational point of view, when the number of spikes present in the system is not bounded.

In this case, many variants of (E)SNP systems were shown to be equal in power with Turing machines:

Already the original model of spiking neural P systems was shown to be computationally complete, i.e., able to generate any recursively enumerable set of non-negative integers (see [10]). From the results established in [1], where extended spiking neural P systems (without astrocytes) were shown to be able to compute any partial recursive function on $\mathbb{N}$, computational completeness can immediately be obtained for ESNP systems with inhibitory axons or astrocytes, too, because just omitting inhibitory axons or astrocytes gives a sufficiently powerful submodel of spiking neural P system as defined in [10].

These results can also be obtained with even more restricted forms of spiking neural P systems, e.g., no time delay (refraction period) is needed, as it was shown in [9].

Although even when using the extended model of spiking neural P systems, the restriction of decaying spikes (i.e., the spikes have a limited lifetime) and/or total spiking (i.e., the whole contents of a neuron is erased when it spikes) do not allow for the generation or the acceptance of more than regular sets of natural numbers as it is shown in [6].

When bounding the number of spikes in the neurons, the power of (E)SNP systems decreases:

A characterization of semilinear sets was obtained by spiking neural P systems ([10]), while it was shown in [1] that only regular functions can be computed and only regular sets can be generated/ accepted with finite extended spiking neural P systems.

We refer to the respective cited papers for more details and proofs as well as to [22] for further results.

5 Further Remarks

We have briefly discussed some variants of (extended) spiking neural P systems and their computational power. Many other variants and aspects have already been explored in this young research area, that are not covered here, e.g., the asynchronous use of rules in [5], a sorting algorithm for vectors of natural numbers was proposed in [11], while the solution of some numerical NP-complete problems is addressed in [12], to mention only a few. For the actual state of research and further references, we refer the interested reader to [22].

However, many problems are still open (some of them are listed in [19]) and remain interesting topics for future research.

References

[1] Alhazov A, Freund R, Oswald M, Slavkovik M (2007) Extended spiking neural P systems generating strings and vectors of non-negative integers, in Hoogeboom HJ, Păun Gh, Rozenberg G (eds.), Membrane Computing, 7th International Workshop, WMC7, Springer, Berlin, 123–134

[2] Binder A, Freund R, Oswald M (2008) Extended spiking neural P systems with astrocytes - Variants for modelling the brain, this volume

[3] Binder A, Freund R, Oswald M, Vock L (2007) Extended Spiking Neural P Systems with Excitatory and Inhibitory Astrocytes, In: Proceedings of 8th WSEAS Conference on Evolutionary Computing, Lecture Notes in Computational Intelligence, Vancouver, Canada, 88–101

[4] Buarque de Lima Neto F, de Wilde P (2006) Venn-like models of neo-cortex patches, International Joint Conference on Neural Networks, 89–96

[5] Cavaliere M, Egecioglu O, Ibarra OH, Woodworth S, Ionescu M, Păun Gh (2007) Asynchronous spiking neural P systems, Technical Report 9/2007, Microsoft Research - University of Trento, Centre for Computational and Systems Biology

[6] Freund R, Ionescu M, Oswald M (2007), Extended spiking neural P systems with decaying spikes and/or total spiking, in Vaszil Gy (ed) Automata for Cellular an Molecular Computing, MTA SZTAKI, Budapest, 64–76

[7] Freund R, Oswald M (2007) Extended spiking neural P systems with inhibitory axons, in Sugisaka M, Tanaka H (eds.), Proc.12th International Symposium on Artificial Life and Robotics AROB12th 2007, Beppu, Oita, Japan, 509–512

[8] Gerstner W, Kistler W (2002) Spiking Neuron Models. Single Neurons, Populations, Plasticity. Cambridge Univ. Press

[9] Ibarra OH, Păun A, Păun Gh, Rodríguez-Patón A, Sosík P, Woodworth S (2006) Normal forms for spiking neural P systems, in: Fourth Brainstorming Week on Membrane Computing, Vol. II, RGNC REPORT 02/2006, Fénix Editora, Sevilla, 105–136

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

[11] Ionescu M, Sburlan D (2007), Some applications of spiking neural P systems, Pre-proc. Membrane Computing, International Workshop (WMC8), Thessaloniki, Greece, 383–394

[12] Leporati A, Zandron C, Ferretti C, Mauri G (2007), Solving numerical NP-complete problems with spiking neural P systems, Pre-proc. Membrane Computing, International Workshop (WMC8), Thessaloniki, Greece, 405–424

[13] Maass W (2002) Computing with spikes. Special Issue on Foundations of Information Processing of TELEMATIK 8, 1:32–36

[14] Maass W, Bishop C (eds) (1999) Pulsed Neural Networks. MIT Press, Cambridge

[15] Martín-Vide C, Pazos J, Păun Gh, Rodríguez-Patón A (2003) Tissue P systems. Theoretical Computer Science 296 (2):, 295–326

[16] Păun Gh (1998) Computing with membranes. Journal of Computer and System Sciences 61(1):108–143 and TUCS Research Report 208 (1998) (`http://www.tucs.fi`)

[17] Păun Gh (2002) Computing with Membranes: An Introduction. Springer, Berlin

[18] Păun Gh (2007) Spiking neural P systems. A tutorial, Bulletin of the EATCS 91, 145–159.

[19] Păun Gh (2007) Twenty Six Research Topics About Spiking Neural P Systems, available at [22].

[20] Perea G Araque A (2002), Communication between astrocytes and neurons: a complex language, Journal of Physiology Paris 96, 199–207.

[21] Shen X, de Wilde P (2007) Long-term neuronal behavior caused by two synaptic modification mechanisms. Neurocomputing 70, 7-9:1482–1488

[22] The P Systems Web Page, `http://psystems.disco.unimib.it`