Wireless Spiking Neural P Systems

Artiom Alhazov   Rudi Freund

Institute of Math. and Computer Science
Chișinău, Moldova
TU Wien, Austria

BWMC 2025

Overview

Wireless Spiking Neural P Systems

Wireless Spiking Neural P Systems first were presented last year at the Brainstorming in Sevilla by David Orellana-Martín.

Then they were presented at the CMC 2024 in Nice:

- David Orellana-Martín, Francis George C. Cabarle, Prithwineel Paul, XiangXiang Zeng, Rudolf Freund:
  Wireless Spiking Neural P Systems.
  CMC 2024, Nice.

Since then, we have improved several results, but many challenges still remain for further investigations.

Wireless Spiking Neural P Systems

Definition

A *WSN P system* of degree $m \geq 1$ is a construct
$\Pi = (O, \sigma_1, \ldots, \sigma_m)$ where:
$O = \{a\}$ is the singleton alphabet (a is called *spike*);
$\sigma_i = (n_i, E_i, R_i), 1 \leq i \leq m$, is a neuron such that:
– $n_i \in \mathbb{N}$ is the *initial number* of spikes in neuron σ_i;
– $E_i \subseteq NFIN(a)$, the *input filter* of neuron σ_i;
– R_i is a finite set of *rules* of two possible forms:

- $E/a^c \rightarrow a^s$ where $E \subseteq NREG(a)$ is a regular set of numbers over O and $c, s, d \in \mathbb{N}, c, s \geq 1$ (*spiking* rules);
- $a^s \rightarrow \lambda$ where $s \in \mathbb{N}, s \geq 1$ (*forgetting* rules);

Wireless Spiking Neural P Systems

A WSN P system $\Pi = (O, \sigma_1, \ldots, \sigma_m)$ of degree $m \geq 1$ can be seen as a a set of m neurons labeled by $1, \ldots, m$ such that:

1. $n_1, \ldots, n_m$ represent the *initial multisets* of objects a (spikes) situated at the beginning in the m neurons of the system;
2. $E_1, \ldots, E_m$ are finite sets over O assigned to the m neurons of the system, working as *input filters* for the spike packages allowed to enter the neuron;
3. $R_1, \ldots, R_m$ are finite sets of rules governing the dynamics of the system.

Computations in Wireless Spiking Neural P Systems

A *configuration* of a WSN P system Π at some moment of time t is described as
$C_t = \langle (n_{1,t}), \ldots, (n_{m,t}) \rangle$
with the number of spikes $n_{i,t}$ in each neuron i.
$C_0 = \langle (n_1), \ldots, (n_m) \rangle$ is the initial configuration.

Applicability of and results of applying rules:
spiking rule $E/a^c \to a^s \in R_i$: in the neuron labeled by i $a^{n_{i,t}} \in E$; c spikes are removed from neuron i s spikes are produced in the environment.

forgetting rule $a^s \to \lambda \in R_i$: the neuron labeled by i contains exactly s spikes; no spikes are generated.

Semantics of Applying Rules

1. *spike packages semantics, pac*: Each package of spikes is treated separately: if $\{a^{c_1}, \ldots, a^{c_k}\}$ is the multiset of packages of spikes produced by neurons that have applied a spiking rule in the current step, only all the neurons σ_i such that $a^{c_j} \in E_i$ receive c_j spikes.

2. *total spikes semantics, tot*: we take the *sum* of all the spikes produced by the neurons of the system $c = \sum_{j=1}^{k} c_j$; all the neurons σ_i with $a^c \in E_i$ receive c spikes.

Semantics of Applying Rules

At some instance t we say the configuration C_t of the WSN P system Π produces a configuration C_{t+1} in one *step* – we denote that by $C_t \Rightarrow_\Pi C_{t+1}$ – by executing the following two substeps:

- *all* neurons apply *one rule* (if possible);
- *each* neuron σ_i according to the underlying semantics $\alpha \in \{pac, tot\}$ takes the (packages of) spikes produced in the first substep from the environment if they can pass the input filter E_i of σ_i.

Semantics of Applying Rules

We assume a *global clock* to *synchronise* the computations in Π, that is, if a neuron can apply a rule then it must do so. In every step Π is *locally sequential* since at most one rule (to be chosen in a nondeterministic way) in each neuron can be applied, but *globally parallel* as more than one neuron can apply a rule.

A *computation* of a WSN P system Π is defined as a (finite or infinite) sequence of configurations $\mathcal{C} = (C_0, C_1, \ldots, C_n, \ldots)$, where C_0 is the initial configuration of Π and $C_t \Rightarrow_\Pi C_{t+1}$ for all t.

If after n steps no more rules as described above can be applied, we say that Π halts after n steps, and $\mathcal{C} = (C_0, C_1, \ldots, C_n)$ is called a *halting computation*.

Remark

We assume the spikes present in the environment to be available for all neurons only for one computation step.

Output

Let $\Pi = (O, \sigma_1, \ldots, \sigma_m)$ be a WSN P system working in the semantics $\alpha \in \{pac, tot\}$.
There are several ways how at the end of a halting computation the output of the system can be obtained:

- The output consists of a k-vector of natural numbers given by the number of spikes in some designated output neurons $\sigma j_1, \ldots, \sigma j_k$; in that case the whole WSN P system is given as

$$\Pi = (O, \sigma_1, \ldots, \sigma_m; j_1, \ldots, j_k)$$

Output

We may also distinguish the following subcases:

$Ps_{\alpha,k-out}(\Pi)$: the numbers in the k-vector are directly given by the number of spikes contained in the output neurons;

$Ps_{\alpha,out_f}(\Pi)$: if the numbers for the output vector are encoded in the number of spikes contained in the output neurons by a specific function f, e.g., an exponential function; if f is a linear function, we write $Ps_{\alpha,k-out_l}(\Pi)$.

Output

- The output is obtained from a designated output neuron σ_{i_0}, $1 \leq i_0 \leq m$; in that case the whole WSN P system is given as

$$\Pi = (O, \sigma_1, \ldots, \sigma_m; i_0),$$

and we may distinguish the following subcases:

$Ps_{\alpha,k}WSNP$: the output vector with k components is obtained as the *time intervals* $\langle t_2 - t_1, \ldots, t_{k+1} - t_k \rangle$;

$Ps_{\alpha,k-sequ}(\Pi)$: the output vector with k components is obtained as k sequences of consecutive spikes sent to the environment by the output neuron σ_{i_0}.

Output

The families of sets of k-vector of natural numbers obtained by WSN P systems as described above are denoted by $Ps_{\alpha,k-out}WSNP$, $Ps_{\alpha,k-out_f}WSNP$, $Ps_{\alpha,k-int}WSNP$, and $Ps_{\alpha,k-sequ}WSNP$.

In all variants, we replace Ps by N, if only one natural number is to be obtained as output.

Remark

In the case with the designated output neuron σ_{i_0} we can think of σ_{i_0} as the interface of Π to the environment. As a technical detail we mention that in contrast to SN P systems, the firing of σ_{i_0} should only send spikes to the environment, but not to any of the neurons in Π including σ_{i_0}. This may be accomplished by avoiding a to be contained in any of the input filters E_i, $1 \leq i \leq m$.

An Optimal Result for WSN P Systems

The following result is optimal with respect to the number of neurons:

Theorem

The computations of any register machine with m registers can be simulated by a WSN P system with m neurons, with the input and output being encoded in a linear way.

The result holds for both spike modes pac, tot.

Proof of the Optimal Result for WSN P Systems

Consider a register machine with m registers

$$M = (m, B, l_0, l_h, P)$$

with $|B| = l = |P|$.
We list instructions and registers as $\langle l_0, l_1, \ldots, l_h \rangle = \langle 1, 2, \ldots, l \rangle$ and $\langle reg_1, \ldots, reg_m \rangle$.
To the elements of the second list $\langle reg_1, \ldots, reg_m \rangle$, we assign odd prime numbers $P(reg_i)$ in such a way that $P(reg_1) < P(reg_2) < \ldots < P(reg_m)$, but in addition we require $4l < 2P(reg_1)$.

Proof of the Optimal Result for WSN P Systems

If a register reg_i contains the number n then the corresponding neuron σ_{reg_i} contains $a^{2P(reg_i)n}$, i.e., $2P(reg_i)n$ spikes; hence, the number n is encoded by the linear function $2P(reg_i)n$.

In addition, for the simulation of the *ADD*- and *SUB*-instructions on this register reg_i of M, σ_{reg_i} may contain an additional odd number of spikes, which is less than the number representing the lowest non-zero value $2P(reg_i)$.

Proof of the Optimal Result for WSN P Systems

The WSN P system Π now is defined as

$$\Pi = (\{a\}, \sigma_{reg_1}, \ldots, \sigma_{reg_m})$$

Defintion of the neurons σ_{reg_i}, $1 \leq i \leq m$:
$\sigma_{reg_i} = (Initial_{reg_i}, R_{reg_i}, E_{reg_i})$.

$Initial_{reg_i} = a^{2P(reg_i)n_i}$
where n_i is the initial value in register i.

$$E_{reg_i} = \{a^{2P(reg_i)}\} \cup \{a^{2(l+p)-1} \mid p \in B_{ADD(i)} \cup B_{SUB(i)}\}.$$

Proof of the Optimal Result for WSN P Systems

$$
\begin{aligned}
R_{reg_i} \;=\; & \{\{a^{2jP(reg_r)+2(l+p)-1} \mid 0 \le j\}/a^{2l} \to a^{2P(reg_r))} \\
& \mid p \in B_{ADD(r)}\} \\
\cup\; & \{a^{2jP(reg_r)+2p-1} \mid 0 \le j\}/a^{2p-1} \to a^{2(l+q(p))-1} \\
& \mid p \in B_{ADD(r)}\} \\
\cup\; & \{a^{2jP(reg_r)+2p-1} \mid 0 \le j\}/a^{2p-1} \to a^{2(l+s(p))-1} \\
& \mid p \in B_{ADD(r)}\} \\
\cup\; & \{\{a^{2jP(reg_r)+2(l+p)-1} \mid 1 \le j\}/ \\
& a^{2(l+p)-1+2P(reg_r)} \to a^{2(l+q(p))-1} \mid p \in B_{SUB(r)}\} \\
\cup\; & \{a^{2(l+p)-1} \to a^{2(l+s(p))-1} \mid p \in B_{SUB(r)}\}
\end{aligned}
$$

Extended Spiking Neural P Systems

- Artiom Alhazov, Rudolf Freund, Marion Oswald, and Marija Slavkovik: Extended Spiking Neural P Systems. Hendrik Jan Hoogeboom, Gheorghe Păun, Grzegorz Rozenberg, and Arto Salomaa (eds.): Membrane Computing, 7th International Workshop, WMC 2006, Leiden, The Netherlands, July 17-21, 2006. Lecture Notes in Computer Science, vol. 4361, pp. 123–134, 2006.
- Sergey Verlan, Rudolf Freund, Artiom Alhazov, Sergiu Ivanov, and Linqiang Pan: A formal framework for spiking neural P systems, J. Membr. Comput., vol. 2/4, pp. 355–368, 2020.

Extended Spiking Neural P Systems and WSN P Systems

In extended spiking neural P systems, a rule which is used may send packages to several neurons, hence, the target neurons are indicated directly.

In WSN P systems, the target neurons for the packages of spikes are chosen by their input filters.

Especially when using packages of spikes, WSN P systems can be interpreted as a special variant of extended spiking neural P systems.

The close relations between these two models have to be investigated further in the future.

Future Research on WSN P Systems

- We have only considered one of the output variants. Especially the variant with encoding the output as sequences of time intervals remains to be implemented.
- As in extended spiking neural P systems, we may consider erasing rules with also allowing regular filters, i.e., of the form $E/a^c \to \lambda$. In that way, neurons can be emptied completely.
- Using other encodings, can we avoid input filters?
- $\ldots$

THANK YOU VERY MUCH!

Y una vez más:

MUCHAS GRACIAS

a nuestros colegas de Sevilla por organizar las sesiones de “brainstorming”.

¡El futuro será brillante y prometedor

si nos volvemos a ver el año 2027

para el brainstorming en Sevilla!

BWMC 2027