Spiking neural P systems with colored spikes and multiple channels in the rules (extended abstract)

Rodrigo Llanes, José M. Sempere[1]

Valencian Research Institute for Artificial Intelligence (VRAIN)
Universitat Politècnica de València
`rodllala@inf.upv.es` `jsempere@dsic.upv.es`

We propose SN P systems that allows the sending/receiving of different kinds of spikes through different channels. Our proposal is related to [5], where the authors proposed the use of different spiking channels to connect the neurons in the network. Observe that the idea of connecting different target neurons in the same rule was initially proposed in [1] where the inclusion of target neurons in the rules is allowed. This idea, and the relationship with multiple channels, was highlighted in [7] where the authors considered this ingredient in the formal framework of SN P systems. The other variant that we have considered in this work is showed in [6], where the authors proposed the use of a non-singleton alphabet to define the spikes of the system.

Here, we combine multiple channels in the neuron rules and different kinds of spikes together. The rules can manage different spikes at every moment, and they can be sent by different channels. This combination of ingedients allows the simulation of other models proposed in the membrane computing research area, such as virus machines [3] and neural-like P systems with plasmids [2].

SN P systems

Definition 1. [4] A spiking neural P system (SN P system, for short) of degree $m \geq 1$ is defined by the following tuple $\Pi = (O, \sigma_1, \sigma_2, \ldots, \sigma_m, syn, in, out)$, where:

1. $O = \{a\}$ is a singleton alphabet of spikes
2. $\sigma_1, \sigma_2, \ldots, \sigma_m$ are neurons of the form $\sigma_i = (n_i, R_i)$, $1 \leq i \leq m$, where
   a) $n_i \geq 0$ is the initial number of spikes contained in σ_i.
   b) R_i is a finite set of rules in one of the following two forms:
      - firing or spiking rules: $E/a^c \to a; d$ where E is a regular expression over O, and $c \geq 1,\ d \geq 0$ are integer numbers. We omit E whenever it be equal to a^c, and we omit d whenever it be equal to 0.

[1] Corresponding author

- forgetting rules: $a^s \to \lambda$, for $s \geq 1$, with the restriction that for each spiking rule $E/a^c \to a; d$ then $a^s \notin L(E)$ ($L(E)$ is the regular language defined by E)

3. $syn \subseteq \{1, 2, \dots m\} \times \{1, 2, \dots, m\}$ with $(i, i) \notin syn$ for $1 \leq i \leq m$, is the directed graph of synapses between neurons
4. $in,\ out \in \{1, \dots, m\}$ indicate the input and the output neurons of Π.

$\square$

Definition 2. [5] A spiking neural P system with multiple channels (SNP-MC system, for short) of degree $m \geq 1$ is defined by $\Pi = (O, L, \sigma_1, \sigma_2, \dots, \sigma_m, syn, out)$, where:

1. $O = \{a\}$ is a singleton alphabet of spikes
2. $L = \{1, 2, \dots, N\}$ is the alphabet of channel labels
3. $\sigma_1, \sigma_2, \dots, \sigma_m$ are neurons of the form $\sigma_i = (n_i, L_i, R_i)$, $1 \leq i \leq m$, where:
   a) $n_i \geq 0$ is the initial number of spikes contained in σ_i.
   b) $L_i \subseteq L$ is a finite set of channels labels used in the neuron
   c) R_i is a finite set of extended rules in the form $E/a^c \to a^p(l)$ where E is a regular expression over O, and $c \geq 1,\ p \geq 0, l \in L_i$.
4. $syn = \{(i, j, l)\} \subseteq \{1, 2, \dots m\} \times \{1, 2, \dots, m\} \times L$ with $(i, i, l) \notin syn$ for $1 \leq i \leq m$ and $l \in L$ (synapse connections);
5. $out \in \{1, \dots, m\}$ is the output neuron.

$\square$

In this case, the forgetting rules can be established as those extended rules with $p = 0$ and no regular expression E. Every rule indicates the channel that the rule uses to send the spikes. The spikes are only sent to the connected neurons by the channel established in the rule. In each computation step there can be some competition between different rules at the neuron. Provided that more than one rule can be applied, then the system selects only one of them non-deterministically. So, the neurons work in sequential manner (they apply only one rule at every computation step).

Definition 3. [6] A spiking neural P system with colored spikes (SNP-CS system) of degree $m \geq 1$ is defined by $\Pi = (C, O, \sigma_1, \sigma_2, \dots, \sigma_m, syn, in, out)$, where:

1. $C = \{1, 2, \dots g\}$ is a finite set of colors to mark the color of a spike. Every spike is associated with a unique color
2. $O = \{a_1, a_2, \dots, a_g\}$ is an alphabet of g colored spikes
3. $\sigma_1, \sigma_2, \dots, \sigma_m$ are neurons of the form $\sigma_i = (< n_1^i, n_2^i, \dots, n_g^i >, R_i)$, $1 \leq i \leq m$, where:
   a) $n_h^i \in \mathbb{N}$ is the number of spikes a_h initially placed in neuron σ_i.
   b) R_i is a finite set of spiking and forgetting rules in the form:
      - Spiking rule: $E/a_1^{c_1} a_2^{c_2} \dots a_g^{c_g} \to a_1^{p_1} a_2^{p_2} \dots a_g^{c_g}; d$ where E is a regular expression over O, and $c_i, p_i \geq 0$;

- Forgetting rule: $a_1^{s_1} a_2^{s_2} \dots a_g^{s_g} \to \lambda$ where $a_1^{s_1} a_2^{s_2} \dots a_g^{s_g} \notin L(E)$ for any regular expression E associated with a spiking rule;

4. $syn \subseteq \{1, 2, \dots m\} \times \{1, 2, \dots, m\}$ with $(i, i) \notin syn$ for $1 \le i \le m$;
5. $in, out \in \{1, \dots m\}$ are the input and the output neurons

□

The work in this model is similar as in the original model described in [4].

Spiking neural P systems with multiple channels in the rules and colored spikes

We introduce a model of SN P systems that combines the use of multiple channels that can be located both in the synaptic connections and in the rules of the neurons themselves, together with a colored spikes alphabet (i.e. a non-singleton alphabet).

Definition 4. A spiking neural P system with multiple channels and colored spikes (SNP-MC-CS system, for short) of degree $m \ge 1$ is defined by $\Pi = (O, L, \sigma_1, \sigma_2, \dots, \sigma_m, syn, in)$, where:

1. $O = \{a_1, a_2, \dots, a_g\}$ is an alphabet with g colored spikes
2. $L = \{1, 2, \dots, N\}$ is an alphabet of channel labels
3. $\sigma_1, \sigma_2, \dots, \sigma_m$ are neurons of the form $\sigma_i = (w_i, L_i, R_i)$, $1 \le i \le m$, where:
   a) $w_i \in O^*$ is a string that denotes the initial multiset of spikes in the neuron.
   b) $L_i \subseteq L$ is a finite set of channels labels used in the neuron
   c) R_i is a finite set of rules in the forms:
      - Firing rules: $E/w_c \to w_1\,(l_1)\,w_2\,(l_2) \dots w_n\,(l_n)\,; d$ where E is a regular expression over O, and $w_c, w_i \in O^*$ $1 \le i \le n, l_i \in L_i$, $d \ge 0$.
      - Forgetting rules: $w_c \to \lambda$, where $w_c \in O^*$.
4. $syn \subseteq \{1, 2, \dots m\} \times \{1, 2, \dots, m, out\} \times L$ are the set of synapse connections with $(i, i, l) \notin syn$ for $1 \le i \le m$ and $\forall\ l \in L$. Observe that (i, out, l) denotes that the neuron σ_i sends the spikes out to the environment by the channel l.
5. $in \in 1, \dots, m$ is the input neuron. Observe that the input neuron can be omitted whenever the system is a generator.

□

The firing rules $E/w_c \to w_1\,(l_1)\,w_2\,(l_2) \dots w_n\,(l_n)\,; d$ of the neuron σ_i can be applied whenever $\psi_O(x_i) \in \Psi_O(L_E)$, where x_i denotes the spikes in the neuron σ_i, $\psi_O(x_i)$ denotes the Parikh set of x_i, and L_E is the language denoted by the regular expression E. In such a case, the spikes denoted by w_c are removed from the neuron, and the spikes denoted by w_i are sent to the connected neurons (or to the environment) by the channel l_i according to the synaptic connections of the neuron. If the delay $d > 0$, then the neuron is blocked, and it cannot neither send or receive spikes after d computation steps.

The forgetting rules of the neuron σ_i in the form $w_c \to \lambda$, can be applied whenever no firing rule is applicable and $(\forall a_i \in O)\ |w_c|_{a_i} \le |x_i|_{a_i}$, where x_i

denotes the spikes in the neuron σ_i. In such a case, the spikes denoted by w_c are removed from the neuron.

The system runs in a maximally parallel manner, that is, at every computation step, every neuron that has some applicable rule will apply it. If there are more than one applicable rule, the firing rules will have priority over those of forgetting. If there are more than one applicable firing rule in conflict (they require common spikes) then, only one of them is non-deterministically selected to be applied, the chosen rule will be executed as many times as possible. Observe that, in this case, the system behavior is similar to the one of transition cell-like P systems. If a rule with a delay is activated, the neuron will be blocked just as in SN P systems. If several rules with delays are applied at the same time, the neuron will be blocked for the maximum delay value of the applied rules. Finally, it should be noted that the compution halts when there is no delayed neuron, and no rule can be applied in any neuron. Regarding the output of the system, we can obtain it in four different ways:

- *halting mode*: The result of the computation is the number of spikes that have been sent to the environment throughout computation, no matter their colors. The output is obtained when the system halts.
- *multi-channel halting mode*: As in the halting mode, the output of the model are the spikes sent to the environment during execution, but they are joined depending on the channels through which they have been sent (i.e. a non-negative integer vector is obtained).
- *temporary mode*: This mode is the one used for generating systems, since it is not necessary reach a halting configuration, the system output will be read as the sequence of spikes sent to the environment at every computation step (a spike train).
- *multi-channel temporary mode*: It can be used for signal processing, as it allows reading the output as the sequence of spikes (a spike train) sent out to the environment at every computation step for every channel.

Theorem 1. SNP-MC-CS systems are universal models of computation.

Simulating other computation models.

The proposed model is able of simulating different models that have been previously proposed in the field of membrane computing. In this way, the SNP-MC-CS model aims to be a unifying framework for other alternative models.

Definition 5. [3] A Virus Machine (VM, for short) of degree $l(p, q)$,$p, q \geq 1$, is a tuple $\Pi = (\Gamma, H, I, D_H, D_I, G_C, n_1, \ldots, n_p, i_1, h_{out})$ where:

1. $\Gamma = v$ is a singleton alphabet;
2. $H = h_1, \ldots, h_p$ and $I = i_1, \ldots, i_q$ are ordered sets such that $v \notin H \cup I$ and $H \cap I = \emptyset$;

3. $D_H = (H \cup \{h_{out}\}, E_H, w_H)$ is a weighted directed graph, where $E_H \subseteq H \times (H \cup \{h_{out}\})$, $(h, h) \notin E_h$ for each $h \in H$, *out-degree*$(h_{out}) = 0$, and w_H is a mapping from E_H onto $\mathbb{N} - \{0\}$ (the set of positive integer numbers);
4. $D_I = (I, E_I, w_I)$ is a weighted directed graph, where $E_I \subseteq I \times I$, w_I is a mapping from E_I onto $\mathbb{N} - \{0\}$ and, for each vertex $i_j \in I$, the out-degree of i_j is less than or equal to 2;
5. $G_C = (V_C, E_C)$ is an undirected bipartite graph, where $V_C = I \cup E_H$, being I, E_H the partition associated with it. In addition, for each vertex $i_j \in I$, the degree of i_j is less than or equal to 1.
6. $n_j \in \mathbb{N}(1 \leq j \leq p)$;
7. $h_{out} \notin I \cup v$ and h_{out} is denoted by h_0 in the case that $h_{out} \notin H$.

□

Theorem 2. For every Virus Machine there exists an equivalent SNP-MC-CS system.

Definition 6.[2] A neural-like P systems with plasmids (NP P system, for short) of degree $m \geq 1$, is a construct of the form $\Pi = (O, b_1, \ldots, b_m, link, in, out)$ where:

1. $O = \{p_1, \ldots, p_w\}$ is an alphabet of *plasmids*
2. $b_1, \ldots, b_m$ are bacteria of the form $b_i = (N_i, R_i)$ for $1 \leq i \leq m$ where $N_i = \langle n_1, n_2, \ldots, n_w \rangle$ is a vector and $n_j \geq 0$ is the initial number of plasmids p_j inside bacterium b_i for $1 \leq j \leq w$; and R_i is a finite non-empty set of plasmid rules of the form $C/r_1, r_2, \ldots, r_n$ where condition C is a multiset over O, and, each r_k for $1 \leq k \leq n$ is either one of two forms:
   a) A *transmit* operation: $P \rightarrow out$, where P is a submultiset of C;
   b) A *kill* operation: $P \rightarrow \lambda$, where P is a submultiset of C;
3. $link \subset \{1, 2, \ldots, m\} \times \{1, 2, \ldots, m\}$ with $(i, i) \notin link$ for $1 \leq i \leq m$ (the links between bacteria);
4. in, out indicates the input and output bacterium of the system, respectively. They can be omitted, depending on whether the system is generating outputs or accepting inputs.

□

Theorem 3. For every NP P system there exists an equivalent SNP-MC-CS system.

Conclusions

In this work we have proposed the combination of two variants that had been previously proposed. We propose that the combination of both variants allows the simulation of very diverse models, so the combined model can be viewed as an unification one.

Our future work will be based on considering other models and checking if our proposal is still so general as to be able to simulate them.

References

1. A. Alhazov, R. Freund, M. Oswald, M. Slavkovik (2006) Extended Spiking Neural P Systems. In: H.J. Hoogeboom, G. Păun, G. Rozenberg, A. Salomaa (eds). Membrane Computing. WMC 2006. LNCS vol, 4361. pp 123-134 Springer.
2. F.G.C. Cabarle, X. Zeng, N. Murphy, T. Song, A. Rodríguez-Patón, X. Liu. (2021) Neural-like P systems with plasmids. *Information and Computation*, Vol. 281, 194776.
3. X. Chen, M.J. Pérez-Jiménez, L. Valencia-Cabrera, B. Wang, X. Zeng (2016) Computing with viruses. *Theoretical Computer Science* 623 pp 146-159.
4. M. Ionescu, Gh. Păun, T. Yokomori (2006) Spiking neural P systems. *Fundamenta Informaticae*, 71 pp 279-308
5. H. Peng, J. Yang, J. Wang, T. Wang, Z. Sun, X. Song, X. Lou. (2017) Spiking neural P systems with multiple channels. *Neural Networks,* Vol. 95, pp 66-71.
6. T. Song, A. Rodríguez-Patón, P. Zheng, X. Zeng (2018) Spiking Neural P Systems With Colored Spikes. *IEEE Transactions on Cognitive and Developmental Systems* 10(4) pp 1106-1115.
7. S. Verlan, R. Freund, A. Alhazov, S. Ivanov, L. Pan (2020) A formal framework for spiking neural P systems. *Journal of Membrane Computing* Vol. 2, pp 355–368.