Fuzzy reasoning spiking neural P systems revisited: A formalization

Mario J. Pérez-Jiménez, Carmen Graciani, David Orellana-Martín, Agustín Riscos-Núñez, Álvaro Romero-Jiménez, Luis Valencia-Cabrera

Research Group on Natural Computing, Universidad de Sevilla, Av. Reina Mercedes, s/n, Sevilla, Spain

Keywords:
Fuzzy knowledge representation
Fuzzy reasoning
Spiking neural P systems
Fault diagnosis

A B S T R A C T

Research interest within membrane computing is becoming increasingly interdisciplinary. In particular, one of the latest applications is fault diagnosis. The underlying mechanism was conceived by bridging spiking neural P systems with fuzzy rule-based reasoning systems. Despite having a number of publications associated with it, this research line still lacks a proper formalization of the foundations.

1. Introduction

Spiking neural P systems (SN P systems), since first presented in 2006 [1], have proved to be versatile devices for several practical applications. More recently, a variant of SN P systems, the so-called *Fuzzy Reasoning Spiking Neural P systems* (FRSN P systems, for short) were defined in [2], providing interesting features that make them suitable for modeling fault diagnosis systems, usually including a number of elements and relations involving different types of uncertainty. These aspects imply the need of handling fuzzy knowledge and reasoning when dealing with fault diagnosis applications.

The mechanisms provided within FRSN P systems to aid in the study of this kind of fault diagnosis problems set a bridge between spiking neural P systems and fuzzy rule-based reasoning systems. However, despite the emergence of a number of publications about this topic [2–4], a more *in-depth* theoretical study was necessary to provide a proper formalization of the foundations.

The present work aims to provide a solid formalization of FRSN P systems. More specifically, it formalizes the first variant presented in [2], dealing with real numbers. Latter variants of these systems might be addressed by future works. The structure of the paper is as follows. First, Section 2 includes some general definitions used in the rest of the text, in order to make this document self-contained. Second, the formalization of FRSN P systems is given in Section 3. Third, Section 4 provides an explanation about the process to build an FRSN P system from a fuzzy reasoning knowledge base. Then, the process is illustrated in Section 5 through a practical example of fault diagnosis expert system from literature. Finally, Section 6 summarizes the main conclusions of this paper, while outlining some possible research lines of future work.

2. Preliminaries

2.1. Fuzzy production systems

Expert systems are computer programs designed to emulate the thinking patterns of an expert. In this paper we restrict our attention to the so-called *production systems*, which employ rules of the type "IF (the data meet certain specified conditions) THEN (perform the specified actions)" (see e.g. [5] for more details). The "IF" part of the rule is called the *antecedent* and the "THEN" part is the *consequent*.

There are many sources of uncertainty in data: imprecision in numerical measurements, uncertainty with respect to facts, ambiguous terms, etc. One way to deal with those uncertainties is representing them by real numbers (usually, but not always, between 0 and 1) expressing in some sense how confident we are about the data. Such numbers are called the *truth values* associated with the data.

Rules in fuzzy production systems are expressed by means of *propositions* involving *linguistic variables*. The antecedent (resp. the consequent) of a rule can be formally seen as a set of *linguistic terms*, each of them having associated a *membership function*, allowing to derive from the values of the variables a truth value for the proposition. For example, the variable `Speed` could consist of the terms `slow`, `medium` and `fast`. Then, these terms could have associated membership functions such that, for a speed of 50, the proposition `Speed is slow` would have a truth value of 0, the proposition `Speed is medium` a truth value of 0.6 and the proposition `Speed is fast` a truth value of 0.25.

Linguistic terms can be affected by *hedges*, which are adjectives or adverbs that modify the truth value of a proposition. For example, the truth value of the proposition `Speed is very fast` would be 0.0625, since we assume that the effect of the hedge `very` is to square the truth value.

In fuzzy logic the truth value of the classical `not`, `and` and `or` connectives can be computed using the *Zadeh operators* as follows (note that other operators are possible):

- Truth value of $\texttt{NOT}\ \varphi$: $1 -$ truth value of φ.
- Truth value of $\varphi\ \texttt{AND}\ \psi$: $\min\{$truth value of φ, truth value of $\psi\}$.
- Truth value of $\varphi\ \texttt{OR}\ \psi$: $\max\{$truth value of φ, truth value of $\psi\}$.

In the literature, IF-THEN rules can be classified in various ways. We recall here the classification into five elementary types given in [5].

- Type 1: *simple fuzzy production rules* of the form

  $\texttt{IF}\ \varphi\ \texttt{THEN}\ \psi$

- Type 2: *composite fuzzy conjunctive rules in the antecedent* (a.k.a. `AND-rules`) of the form

  $\texttt{IF}\ \varphi_1\ \texttt{AND}\ \cdots\ \texttt{AND}\ \varphi_n\ \texttt{THEN}\ \psi$

- Type 3: *composite fuzzy conjunctive rules in the consequent* of the form

  $\texttt{IF}\ \psi\ \texttt{THEN}\ \varphi_1\ \texttt{AND}\ \cdots\ \texttt{AND}\ \varphi_n$

- Type 4: *composite fuzzy disjunctive rules in the antecedent* (a.k.a. `OR-rules`) of the form

  $\texttt{IF}\ \varphi_1\ \texttt{OR}\ \cdots\ \texttt{OR}\ \varphi_n\ \texttt{THEN}\ \psi$

- Type 5: *composite fuzzy disjunctive rules in the consequent* of the form

  $\texttt{IF}\ \psi\ \texttt{THEN}\ \varphi_1\ \texttt{OR}\ \cdots\ \texttt{OR}\ \varphi_n$

where each φ_i and ψ are propositions involving linguistic variables.

The initial motivation for this paper is to present a formalization for the seminal definition of FRSN P systems ([2]). This is a deterministic model, using real numbers to represent truth values associated with propositions. Therefore, rules of type 5 fall out of the scope of the formalization presented here, since it is not straightforward to capture their effect in this context.

In what follows, for the sake of simplicity and without loss of generality, we will only refer to two kinds of fuzzy production rules, namely `AND-rules` and `OR-rules`. Note that, on the one hand, elementary rules of type 1 can be seen as a particular case of an `AND-rule`. On the other hand, any rule of type 3 can be replaced by a set of elementary rules (one for each proposition in the consequent), and the resulting system will be equivalent from a reasoning point of view.

To account for the uncertainty we may have about the correctness of the implications represented by the rules, we associate with them *confidence factors*, which are real numbers between 0 and 1. We can then make a process of *inference* to derive truth values from available data as follows: given a rule with confidence factor τ and with truth value α_i for each proposition φ_i in its antecedent,

- If the rule is an `AND-rule`, then the truth value derived for ψ is

$$\min\{\alpha_1, \ldots, \alpha_n\} \cdot \tau$$

- If the rule is an `OR-rule`, then the truth value derived for ψ is

$$\max\{\alpha_1, \ldots, \alpha_n\} \cdot \tau$$

Fuzzy production systems are *data-driven*, that is, all rules compatible with the available data are applied at the same time, instead of sequentially. This means that the final truth value for proposition ψ should be computed as

$$\max\{\alpha_{r_1}, \ldots, \alpha_{r_m}\}$$

where $r_1, \ldots, r_m$ are all the rules with consequent ψ and $\alpha_{r_1}, \ldots, \alpha_{r_m}$ are the truth values for ψ derived from them.

2.2. Spiking neural P systems

In this section we briefly present a variant of computing models in Membrane Computing called *Spiking neural P systems* (SN P systems, for short) introduced by Ionescu et al. [1] (for further details see also [6]). This variant incorporates ideas of spiking neurons, a promising research line in Neural Computing [7,8].

Spiking neural P systems mimic the way that neurons communicate with each other by means of short electrical impulses, identical in shape (voltage), but emitted at precise moments of time. The underlying structure of these systems is a directed graph whose nodes represent neurons, and arcs represent synapses. The impulses are described by the multiplicity of an object a from a singleton alphabet. Object a is called *spike* and it abstracts a quantum of energy. Neurons can send spikes along its outgoing synapses according to a protocol: depending on their current number of spikes, the neurons either fire sending impulses (a certain number of spikes) to the neighboring neurons, or forget the spikes they have. There is a distinguished neuron called output neuron, whose outdegree is 0, that can send spikes to the environment. Consequently, in the environment we get a sequence of spikes, leaving the system at specific moments of time.

A *spiking neural P system* of degree $q \geq 1$, is a tuple $(O, syn, \sigma_1, \ldots, \sigma_q, i_{out})$, where:

- $O = \{a\}$ is the singleton alphabet;
- $syn = (V, E)$ is a directed graph such that $V = \{\sigma_1, \ldots, \sigma_q\}$ and $(\sigma_i, \sigma_i) \notin E$ for $1 \leq i \leq q$;
- $\sigma_i, 1 \leq i \leq q$, is of the form $\sigma_i = (n_i, \mathcal{R}_i)$ where $n_i \geq 0$ and $\mathcal{R}_i$ is a finite set of *rules* of the following two forms:
  (1) $E/a^c \to a^{c'}; d$, being E a regular expression over $\{a\}$, $c \geq c' \geq 1$, and $d \geq 0$ (*firing rules*);
  (2) $a^s \to \lambda$, for some $s \geq 1$, with the restriction that for each rule $E/a^c \to a^{c'}; d$ of type (1) from $\mathcal{R}_i$, we have $a^s \notin L(E)$ (*forgetting rules*);
- $i_{out} \in \{1, 2, \ldots, q\}$ such that $outdegree(i_{out}) = 0$.

A spiking neural P system of degree $q \geq 1$ can be viewed as a set of q neurons $\{\sigma_1, \ldots, \sigma_q\}$ interconnected by the arcs of a directed graph syn, called *synapse graph*. There is a distinguished neuron i_{out}, called output neuron, which communicates with the environment.

If a neuron σ_i contains k spikes at an instant t, and $a^k \in L(E), k \geq c$, then the rule $E/a^c \to a; d$ can be applied. By the application of that rule c spikes are removed from neuron σ_i and the neuron fires producing c' spikes after d time units. If $d = 0$, then the spikes are emitted immediately. If the rule is used in step t and $d \geq 1$, then in steps $t, t+1, t+2, \ldots, t+d-1$ the neuron is *closed*, so that it cannot receive new spikes. In step $t+d$, the neuron spikes and becomes again open, so that it can receive spikes (which can be used in step $t+d+1$). The spikes produced by a neuron σ_i are received for all *open* neuron σ_j such that $(\sigma_i, \sigma_j) \in E$. If σ_i is the output neuron then the spikes are sent to the environment.

The rules of type (2) are *forgetting* rules, and they are applied as follows: If neuron σ_i contains exactly s spikes, then the rule $a^s \to \lambda$ from R_i can be applied. By the application of this rule all s spikes are removed from σ_i.

In spiking neural P systems, a global clock is assumed, marking the time for the whole system. There exist a number of variants of these systems with different syntactic and semantic ingredients, but their detailed description is out of the scope of this paper. For those interested in their study an overview is provided at [9].

3. Formalization of FRSN P systems

Definition 1. A fuzzy reasoning spiking neural P system (FRSN P system, for short) of degree $q \geq 1$ is a tuple

$$\Pi = (A, \sigma_1, \ldots, \sigma_q, syn, IN, OUT, PN, RN_{\mathrm{AND}}, RN_{\mathrm{OR}}, P, C)$$

where:

- $A = \{a\}$ is a singleton alphabet.

- (V, syn) is a directed graph being $V = \{\sigma_1, \ldots, \sigma_q\}$ such that

$$\forall \sigma \in V\ (indegree(\sigma) > 0 \vee outdegree(\sigma) > 0)$$

each arc in syn has a function γ from $[0, 1]$ onto $[0, 1]$ associated with it.
- IN, OUT are the nonempty subsets of V: $IN = \{\sigma \in PN \mid indegree(\sigma) = 0\}$ and $OUT = \{\sigma \in PN \mid outdegree(\sigma) = 0\}$.
- $\{PN, RN_{\mathrm{AND}}, RN_{\mathrm{OR}}\}$ is a partition of the set V such that

$$\forall (\sigma, \sigma') \in syn\ [(\sigma \in PN \wedge \sigma' \in RN) \vee (\sigma \in RN \wedge \sigma' \in PN)]$$

being $RN = RN_{\mathrm{AND}} \cup RN_{\mathrm{OR}}$.
- P, C are tuples of real numbers: $P = (p_1, \ldots, p_q)$ and $C = (c_1, \ldots, c_q)$, with $p_j, c_j \in [0, 1]$, $1 \le j \le q$, and $c_i > 0$ if $\sigma_i \in RN$.
- Each node $\sigma_j \in V$ is a triple $(\alpha_{\sigma_j}, c_j, F_{\sigma_j})$ such that
  - $\alpha_{\sigma_j} \in [0, 1]$.
  - If $\sigma_j \in IN$ then F_{σ_j} is the identity function on $[0, 1]$.
  - If $indegree(\sigma_j) = n_{\sigma_j} > 0$ then F_{σ_j} is the total function from $[0, 1]^{n_{\sigma_j}}$ onto $[0, 1]$ defined as follows:

$$F_{\sigma_j}(x_1, \ldots, x_{n_{\sigma_j}}) = \begin{cases} max\{\gamma_1(x_1), \ldots, \gamma_{n_{\sigma_j}}(x_{n_{\sigma_j}})\}, & \text{if } \sigma_j \in PN \\ max\{\gamma_1(x_1), \ldots, \gamma_{n_{\sigma_j}}(x_{n_{\sigma_j}})\} \cdot c_j, & \text{if } \sigma_j \in RN_{\mathrm{OR}} \\ min\{\gamma_1(x_1), \ldots, \gamma_{n_{\sigma_j}}(x_{n_{\sigma_j}})\} \cdot c_j, & \text{if } \sigma_j \in RN_{\mathrm{AND}} \end{cases}$$

  where γ_i is the function associated with the i-th incoming arc of σ_j.

A fuzzy reasoning spiking neural P system of degree $q \ge 1$ can be viewed as a set of q neurons $\sigma_1, \ldots, \sigma_q$ arranged in a structure given by a directed graph (V, syn) (called *synapse graph*) such that $V = \{\sigma_1, \ldots, \sigma_q\}$. There are two types of neurons: *proposition neurons* (neurons in the set PN) and *rule neurons* (neurons in the set RN). Among the proposition neurons, some of them belong to two distinguished types: those receiving no spikes (*input neurons*) and those sending no spikes (*output neurons*). With respect to rule neurons, we partition them into *AND-type rule neurons* (neurons in the set RN_{AND}) and *OR-type rule neurons* (neurons in the set RN_{OR}). Each synapse (arc of the synapse graph) connects either a proposition neuron with a rule neuron or vice versa, and the function associated with it indicates whether the proposition appears in the rule as a positive or negative literal. More precisely, the associated function is $\gamma(x) = x$ for the positive case, and $\gamma(x) = 1 - x$ for the negative case. For each neuron σ we consider the sets $Presyn(\sigma) = \{\sigma' \in V \mid (\sigma', \sigma) \in syn\}$ and $Postsyn(\sigma) = \{\sigma' \in V \mid (\sigma, \sigma') \in syn\}$.

The system handles information by exciting neurons which fire sending "electrical impulses" (identical in shape, called *spikes*, denoted by the symbol a of the singleton alphabet) to their *post-synaptic* neurons according to specific mechanisms. Each neuron σ_j has three parameters $\alpha_{\sigma_j}, c_j, F_{\sigma_j}$ associated with it, where

- $\alpha_{\sigma_j} \in [0, 1]$ represents the fuzzy potential/pulse value of the spike contained in neuron σ_j.
- $c_j \in [0, 1]$ represents either the fuzzy truth value (in the case $\sigma_j \in PN$) or the confidence factor (in the case $\sigma_j \in RN$).
- F_{σ_j} is a function introduced in the previous definition. This function allows us to compute the pulse value of the spike to be sent out by means of the synapses.

Definition 2. Let σ be a neuron of a FRSN P system. The state of σ at instant $t \ge 0$, denoted by $s(\sigma, t)$, is a tuple $(\alpha_\sigma(t), f_\sigma(t), G_\sigma(t))$ where:

- $\alpha_\sigma(t)$ is a real number in $[0, 1]$ representing the (potential) pulse value of the spike contained in neuron σ at instant t.
- $f_\sigma(t)$ is a Boolean value that represents if the neuron at instant t is ready to fire.
- $G_\sigma(t)$ is a real number in $[0, 1]$ that represents the pulse value of the spike that neuron σ will send along its outgoing synapses at instant t.

Definition 3. Let $\Pi = (A, \sigma_1, \ldots, \sigma_q, syn, IN, OUT, PN, RN_{\mathrm{AND}}, RN_{\mathrm{OR}}, P, C)$ be a fuzzy reasoning spiking neural P system of degree $q \ge 1$.

- A configuration C_t of Π at moment of time $t \ge 0$ is the tuple $(s(\sigma_1, t), \ldots, s(\sigma_q, t))$, where $s(\sigma_j, t)$ denotes the state of neuron σ_j at instant t.
- The initial configuration of Π is the tuple $C_0 = (s(\sigma_1, 0), \ldots, s(\sigma_q, 0))$, where

$$s(\sigma_j, 0) = \begin{cases} (p_j, 1, p_j), & \text{if } \sigma_j \in IN \\ (p_j, 0, 0), & \text{if } \sigma_j \in V \setminus IN \end{cases}$$

Definition 4. For each $t \geq 0$, configuration C_t yields configuration C_{t+1} in one transition step if the following holds:

- If $\sigma_j \in IN$ then $s(\sigma_j, t+1) = \begin{cases} (p_j, 1, p_j) & \text{if } t < t_D \\ (p_j, 0, 0) & \text{if } t \geq t_D \end{cases}$
  where t_D is the maximum length of a simple path from a neuron in IN to a neuron in OUT.
- If $\sigma_j \in V \setminus IN$ then

$$\alpha_{\sigma_j}(t+1) = \begin{cases} max\{\gamma'(G_{\sigma'}(t)) \mid \sigma' \in Presyn(\sigma_j) \wedge f_{\sigma'}(t) = 1\}, & \text{if } \sigma_j \in PN \cup RN_{\mathrm{OR}} \\ min\{\gamma'(G_{\sigma'}(t)) \mid \sigma' \in Presyn(\sigma_j) \wedge f_{\sigma'}(t) = 1\}, & \text{if } \sigma_j \in RN_{\mathrm{AND}} \end{cases}$$

  where γ' denotes the function associated with the arc (σ', σ_j), for each $\sigma' \in Presyn(\sigma_j)$.
  $f_{\sigma_j}(t+1) = 1$ if and only if $\sigma_j \notin OUT$ and $f_{\sigma'}(t) = 1$, for all $\sigma' \in Presyn(\sigma_j)$.

$$G_{\sigma_j}(t+1) = \begin{cases} 0, & \text{if } \ f_{\sigma_j}(t+1) = 0 \\ F_{\sigma_j}(G_{\sigma_j^1}(t), \ldots, G_{\sigma_j^{n_{\sigma_j}}}(t)), & \text{if } \ f_{\sigma_j}(t+1) = 1 \end{cases}$$

  being $Presyn(\sigma_j) = \{\sigma_j^1, \ldots, \sigma_j^{n_{\sigma_j}}\}$.

Note that a neuron $\sigma \in V \setminus IN$ cannot fire unless all neurons in $Presyn(\sigma)$ have simultaneously fired in the previous step. If we allow that neurons may receive inputs from branches having different depths (w.r.t. the neurons in IN), then at some point one of the incoming arcs will provide its spike, while another arc will remain idle, because the spike traveling through the corresponding branch still needs some more steps to reach such arc. Instead of making the neuron wait until all input spikes are collected (removing the condition of simultaneous firing), we propose an alternative: input neurons will keep spiking repeatedly during a number t_D of steps, until we can guarantee that all neurons in the graph have the chance to receive their message.

A *computation* of a fuzzy reasoning spiking neural P system Π is a (finite or infinite) sequence of configurations such that: (a) the first term of the sequence is the initial configuration C_0 of the system; (b) each non-initial configuration of the sequence is obtained from the previous configuration by applying one transition step; and (c) if the sequence is finite with $n+1$ terms (called *halting computation*) then the last term C_n of the sequence is a *halting configuration*, that is, a configuration such that $f_{\sigma_j}(n) = 0$, for each j, $1 \leq j \leq q$.

In such systems, all computations start from an initial configuration and proceed as stated above; only halting computations give a result, which is encoded by the pulse values of the spikes contained in the output neurons at the last step of the computation (in the halting configuration).

4. FRSN P systems based on fuzzy production systems

This section is devoted to explain how to build an FRSN P system for a given fuzzy reasoning case study, assuming we already have a fuzzy production system designed for it. Let $\{r_1, \ldots, r_{n_r}\}$ be the set of rules of the production system (composed of `AND-rules` and `OR-rules`), and let $\{\varphi_1, \ldots, \varphi_{n_p}\}$ be the set of propositions present in the production system (i.e. at either the antecedent or the consequent of any rule).

We shall describe next how to build the corresponding FRSN P system of degree $q = n_p + n_r$, $\Pi = (A, \sigma_1, \ldots, \sigma_q, syn, IN, OUT, PN, RN_{\mathrm{AND}}, RN_{\mathrm{OR}}, P, C)$.

- The symbol representing the spike is defined as usual, $A = \{a\}$.
- We include the following neurons in Π:
  - One neuron σ_j for each proposition φ_j, $1 \leq j \leq n_p$.
  - One neuron σ_{n_p+i} associated with each rule r_i of the production system, $1 \leq i \leq n_r$.
- We include the following arcs in syn, for each rule r_i:
  - $(\sigma_j, \sigma_{n_p+i})$, for each proposition φ_j included in the antecedent.
  - $(\sigma_{n_p+i}, \sigma_k)$, for the proposition φ_k included in the consequent.
- $PN = \{\sigma_1, \ldots, \sigma_{n_p}\}$, $RN_{\mathrm{AND}} = \{\sigma_{n_p+j} \mid r_j$ is an `AND-rule`$\}$ and $RN_{\mathrm{OR}} = \{\sigma_{n_p+j} \mid r_j$ is an `OR-rule`$\}$.
- The sets IN and OUT are defined according to the arcs in syn. $IN = \{\sigma \in PN \mid indegree(\sigma) = 0\}$ and $OUT = \{\sigma \in PN \mid outdegree(\sigma) = 0\}$. Note that, by definition, every proposition in $\{\varphi_1, \ldots, \varphi_{n_p}\}$ appears in at least one rule. Therefore, the condition $\forall \sigma \in V (indegree(\sigma) > 0 \vee outdegree(\sigma) > 0)$ is satisfied.
- P is defined as follows:
  - p_j is the initial truth value of proposition φ_j, for $1 \leq j \leq n_p$.
  - $p_i = 0$, for every $i > n_p$.

- C is defined as follows:
  - $c_j = p_j$, for $1 \leq j \leq n_p$.
  - c_{n_p+i} is the confidence factor associated with rule r_i.

Let us illustrate the above described method with a simple example.

Example 1. Consider a fuzzy production rule, R, of the form

$$\text{IF } \varphi_1 \text{ AND } \ldots \text{ AND } \varphi_k \text{ THEN } \varphi_{k+1} (\text{CF } = \tau)$$

where $\tau \in [0, 1]$ is its confidence factor, $\varphi_1, \ldots, \varphi_{k+1}$ are propositions. Suppose that the truth value of proposition φ_i is p_i, for $1 \leq i \leq k$, then the truth value of proposition φ_{k+1} is evaluated as $min\{p_1, \ldots, p_k\} \cdot \tau$. Such a rule can be modeled by the following FRSN P system of degree $k + 2$

$$\Pi_R = (A, \sigma_1, \ldots, \sigma_k, \sigma_{k+1}, \sigma_R, syn, IN, OUT, PN, RN_{\text{AND}}, RN_{\text{OR}}, P, C)$$

where

- $A = \{a\}$.
- $syn = \{(\sigma_i, \sigma_R)) \mid 1 \leq i \leq k\} \cup \{(\sigma_R, \sigma_{k+1}\}$.
- $IN = \{\sigma_i \mid 1 \leq i \leq k\}$ and $OUT = \{\sigma_{k+1}\}$.
- $PN = \{\sigma_i \mid 1 \leq i \leq k+1\}$, $RN_{\text{AND}} = \{\sigma_R\}$ and $RN_{\text{OR}} = \emptyset$.
- $P = (p_1, \ldots, p_k, 0, 0)$ and $C = (0, \ldots, 0, \tau)$.

It is easy to check that the system is defined according to Definition 1. The maximum length of a path is $t_D = 2$.

The initial configuration of Π is the tuple

$$C_0 = ((p_1, 1, p_1), \ldots, (p_k, 1, p_k), (0, 0, 0), (0, 0, 0))$$

At $t = 1$ configuration C_0 yields configuration

$$C_1 = ((p_1, 1, p_1), \ldots, (p_k, 1, p_k), (0, 0, 0), (\alpha_{\sigma_R}, 1, \alpha_{\sigma_R} \cdot \tau))$$

where $\alpha_{\sigma_R} = min\{p_1, \ldots, p_k\}$.

Computation proceeds as follows

$$C_2 = ((p_1, 1, p_1), \ldots, (p_k, 1, p_k), (\alpha_{\sigma_R} \cdot \tau, 0, 0), (\alpha_{\sigma_R}, 1, \alpha_{\sigma_R} \cdot \tau))$$

$$C_3 = ((p_1, 0, 0), \ldots, (p_k, 0, 0), (\alpha_{\sigma_R} \cdot \tau, 0, 0), (\alpha_{\sigma_R}, 1, \alpha_{\sigma_R} \cdot \tau))$$

$$C_4 = ((p_1, 0, 0), \ldots, (p_k, 0, 0), (\alpha_{\sigma_R} \cdot \tau, 0, 0), (\alpha_{\sigma_R}, 0, 0))$$

being the last one a halting configuration. The result obtained by the system is $\alpha_{\sigma_R} \cdot \tau = min\{p_1, \ldots, p_k\} \cdot \tau$, the value of the spike in the output neuron, as expected.

5. Application on fault diagnosis

This section illustrates with an example how the method given above can be generalized in order to obtain a FRSN P system to model a general fault diagnosis expert system.

Let us consider the expert system given in [10] which is composed by the following propositions. For some of them the corresponding initial truth value is known.

φ_1: Cross section area of turbine's path is too large, $p_1 = 0.6$.
φ_2: Efficiency of assembling unit is too low, $p_2 = 1$.
φ_3: Ventilation side of the guider's blade of turbine wears and tears.
φ_4: Inlet gas temperature of turbine is too low, $p_4 = 0$.
φ_5: Pressurization ratio of the compressor is too low, $p_5 = 0.2$.
φ_6: Flow path of the combustor wears and tears.
φ_7: Flow rate of the fuel in the combustor is too high, $p_7 = 0$.
φ_8: Higher pressure level's spray head of the turbine is broken.
φ_9: Outlet gas temperature of turbine is too high, $p_9 = 0.2$.
φ_{10}: Efficiency of turbine is too low, $p_{10} = 0.9$.
φ_{11}: Flow coefficient of turbine is too low, $p_{11} = 0$.
φ_{12}: Blade of the turbine scales.
φ_{13}: Power of assembling unit is too low, $p_{13} = 0.8$.
φ_{14}: Blade of the turbine wears and tears.
φ_{15}: Inlet gas temperature of turbine is too high, $p_{15} = 0.2$.

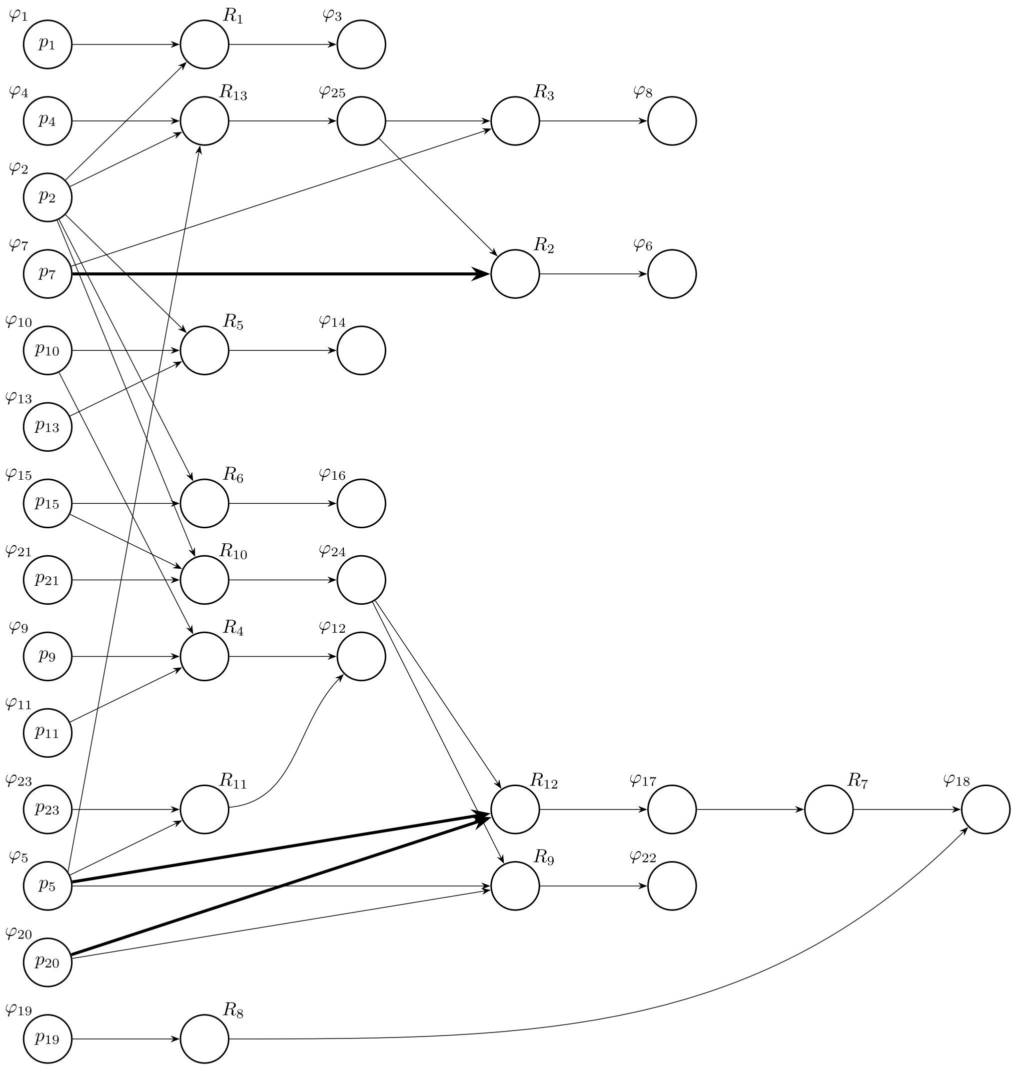

Fig. 1. Example of an FRSN P system. The initial available truth values associated with the propositions are indicated inside the corresponding IN neurons while the rest of neurons are shown empty. Arcs marked in bold have $\gamma(x) = 1 - x$ as their associated function, i.e. they correspond to negative literals.

φ_{16}: Blade of the turbine burns down.
φ_{17}: Flow path of compressor wears and tears.
φ_{18}: Compressor is in turbulence.
φ_{19}: Blade of compressor breaks down, $p_{19} = 0$.
φ_{20}: Conversion flow of the compressor is too low, $p_{20} = 0$.
φ_{21}: Fuel consumption of assembling unit is too high, $p_{21} = 0.3$.
φ_{22}: Inlet of compressor freezes.
φ_{23}: Uniform entropy compression efficiency of compressor is too low, $p_{23} = 0$.
φ_{24}: Compressor has a problem.
φ_{25}: Spray head of turbine is broken.

In order to estimate the unknown truth values the expert system relies on the following fuzzy production rules, given with their corresponding confidence factors.

R_1: IF φ_1 AND φ_2 THEN φ_3, $\tau_1 = 0.8$.
R_2: IF φ_{25} AND $\neg\varphi_7$ THEN φ_6, $\tau_2 = 0.8$.
R_3: IF φ_{25} AND φ_7 THEN φ_8, $\tau_3 = 0.8$.
R_4: IF φ_9 AND φ_{10} AND φ_{11} THEN φ_{12}, $\tau_4 = 0.8$.
R_5: IF φ_{10} AND φ_{13} AND φ_2 THEN φ_{14}, $\tau_5 = 0.8$.

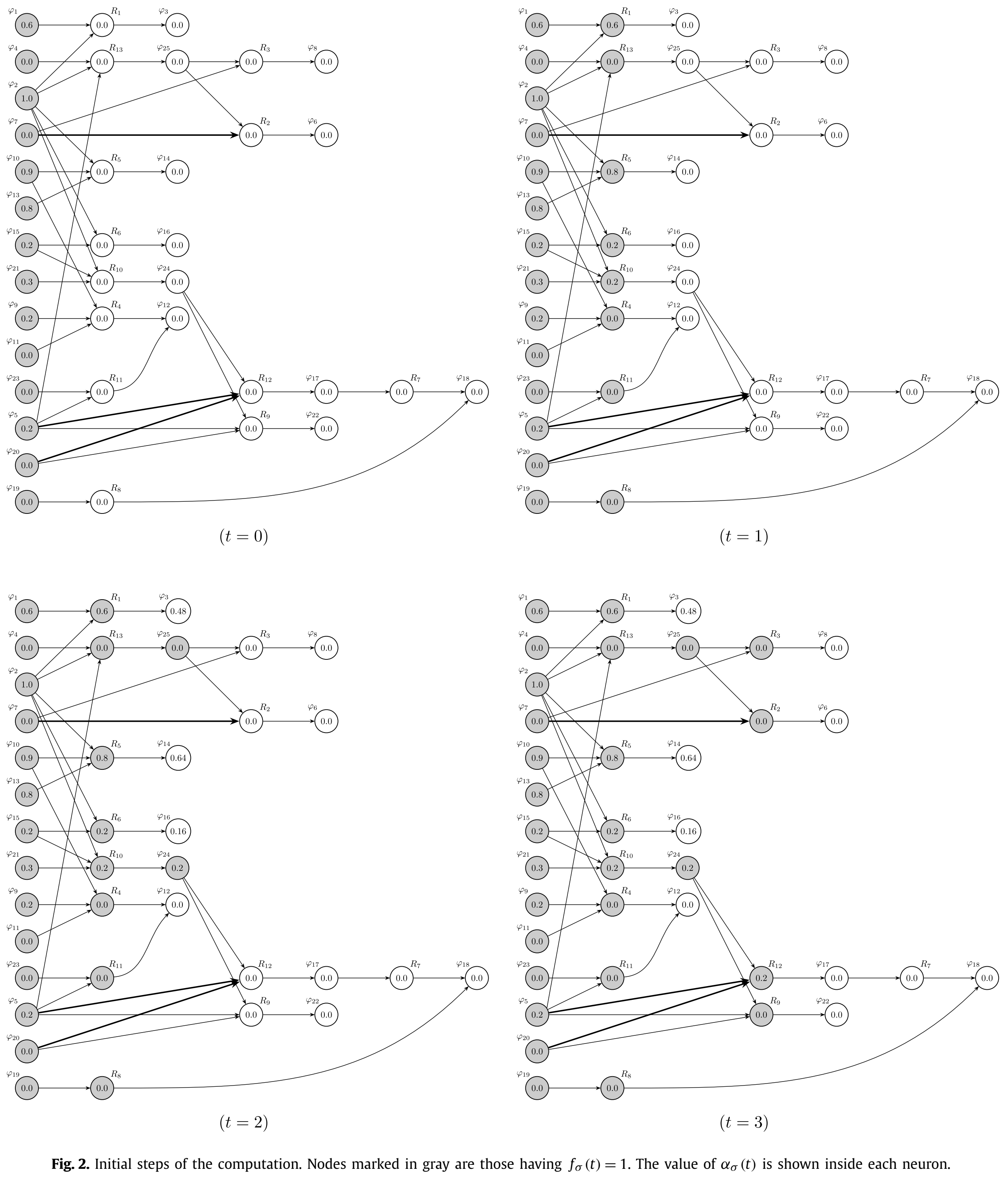

Fig. 2. Initial steps of the computation. Nodes marked in gray are those having $f_\sigma(t)=1$. The value of $\alpha_\sigma(t)$ is shown inside each neuron.

R_6: IF φ_2 AND φ_{15} THEN φ_{16}, $\tau_6=0.8$.
R_7: IF φ_{17} THEN φ_{18}, $\tau_7=0.9$.
R_8: IF φ_{19} THEN φ_{18}, $\tau_8=1$.
R_9: IF φ_{20} AND φ_5 AND φ_{24} THEN φ_{22}, $\tau_9=0.8$.
R_{10}: IF φ_2 AND φ_{21} AND φ_{15} THEN φ_{24}, $\tau_{10}=1$.
R_{11}: IF φ_5 AND φ_{23} THEN φ_{12}, $\tau_{11}=0.9$.
R_{12}: IF $\neg\varphi_{20}$ AND $\neg\varphi_5$ AND φ_{24} THEN φ_{17}, $\tau_{12}=0.8$.
R_{13}: IF φ_4 AND φ_2 AND φ_5 THEN φ_{25}, $\tau_{13}=0.8$.

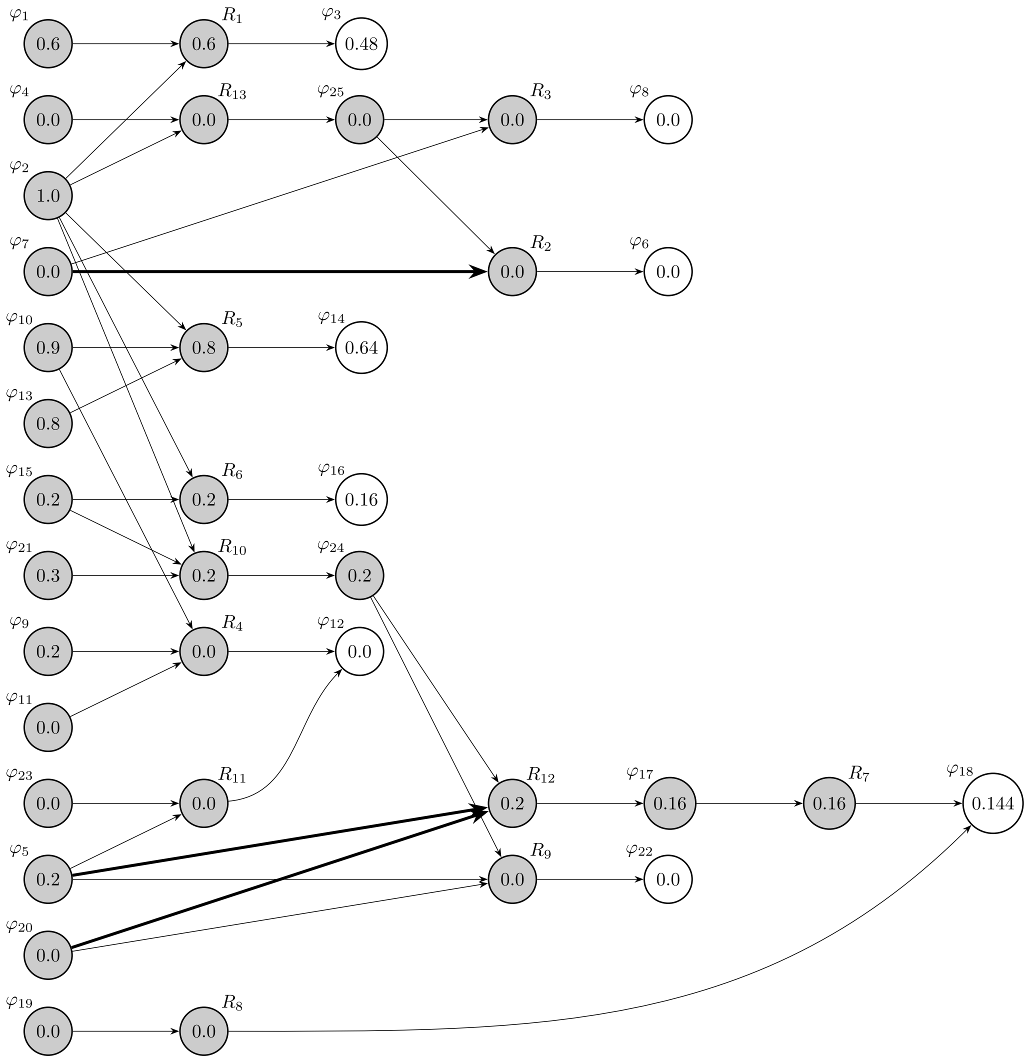

Fig. 3. Configuration at instant $t = 6$.

This fuzzy production system can be modeled by the following FRSN P system of degree 38:

$$\Pi = (A, \sigma_1, \dots, \sigma_{38}, syn, IN, OUT, PN, RN_{\text{AND}}, RN_{\text{OR}}, P, C)$$

- $A = \{a\}$.
- The neurons of the system are:
  - $\sigma_1, \dots, \sigma_{25}$ associated with propositions $\varphi_1, \dots, \varphi_{25}$.
  - $\sigma_{26}, \dots, \sigma_{38}$ associated with rules $R_1, \dots, R_{13}$.
- The arcs in syn are depicted in Fig. 1.
- $PN = \{\sigma_1, \dots, \sigma_{25}\}$, $RN_{\text{AND}} = \{\sigma_{26}, \dots, \sigma_{38}\}$ and $RN_{\text{OR}} = \emptyset$.
  In Fig. 1, PN nodes are labeled with their corresponding proposition and RN nodes are labeled with their corresponding rule.
- $IN = \{\sigma_1, \sigma_2, \sigma_4, \sigma_5, \sigma_7, \sigma_9, \sigma_{10}, \sigma_{11}, \sigma_{13}, \sigma_{15}, \sigma_{19}, \sigma_{20}, \sigma_{21}, \sigma_{23}\}$ and $OUT = \{\sigma_3, \sigma_6, \sigma_8, \sigma_{12}, \sigma_{14}, \sigma_{16}, \sigma_{22}\}$.
- $P = (p_1, \dots, p_{38})$ where p_i is the initial known truth value (if available) and 0 otherwise. $C = (0, \dots, 0, \tau_1, \dots, \tau_{13})$.

The initial steps of the computation are illustrated in Fig. 2. After 6 steps (see configuration C_6 in Fig. 3), all neurons have processed all their inputs, and the estimation of the truth values of all propositions in OUT have been updated. Since $t_D = 6$ for this graph, in the next step IN neurons will stop spiking, and the system will halt in four more steps, when all neurons get their $f_\sigma(t) = 0$. However, $\alpha_\sigma(t)$ values will not change anymore.

In the last configuration, the output neuron σ with the highest α_σ value is σ_{14} ($\alpha_{\sigma_{14}} = 0.64$), so its corresponding proposition $\varphi_{14} \equiv$ 'Blade of the turbine wears and tears' is the most likely fault.

6. Conclusions

This work has provided a complete formalization of FRSN P systems, as initially presented in [2], and illustrated its use through a practical application to a fault diagnosis system.

The chosen example presented in Section 5 includes several interesting features:

- Some propositions appear as negative literals on the rules.
- There exist paths of different lengths connecting IN and OUT neurons.
- Initial truth values 0.0 are allowed for neurons in IN.

Future studies could be conducted to properly formalize other variants of FRSN P systems possibly dealing with different approaches, as trapezoidal numbers instead of real numbers or additional ingredients as weights, among others. We believe that this can be achieved by adjusting the functions F (associated with each neuron) and γ (associated with each arc). Hence, no major change to the formalization presented in this paper seems to be required.

Acknowledgements

This work was partially supported by Grant numbers 61472328 and 61320106005 of the National Natural Science Foundation of China.

References

[1] M. Ionescu, Gh. Păun, T. Yokomori, Spiking neural P systems, Fund. Inform. 71 (2–3) (2006) 279–308.
[2] H. Peng, J. Wang, M.J. Pérez-Jiménez, H. Wang, J. Shao, T. Wang, Fuzzy reasoning spiking neural P system for fault diagnosis, in: Data-Based Control, Decision, Scheduling and Fault Diagnostics, Inform. Sci. 235 (2013) 106–116, http://dx.doi.org/10.1016/j.ins.2012.07.015.
[3] T. Wang, G. Zhang, M.J. Pérez-Jiménez, J. Cheng, Weighted fuzzy reasoning spiking neural P systems: application to fault diagnosis in traction power supply systems of high-speed railways, J. Comput. Theor. Nanosci. 12 (7) (2015) 1103–1114, http://dx.doi.org/10.1166/jctn.2015.3857.
[4] T. Wang, G. Zhang, M.J. Pérez-Jiménez, Fuzzy membrane computing: theory and applications, Int. J. Comput. Commun. Control 10 (6) (2015) 904–935, http://dx.doi.org/10.15837/ijccc.2015.6.2080.
[5] W. Siler, J.J. Buckley, Fuzzy Expert Systems and Fuzzy Reasoning, Wiley–Interscience, 2005.
[6] Gh. Păun, G. Rozenberg, A. Salomaa (Eds.), The Oxford Handbook of Membrane Computing, Oxford University Press, 2010.
[7] W. Gerstner, W. Kistler, Spiking Neuron Models. Single Neurons, Populations, Plasticity, Cambridge University Press, 2002.
[8] W. Maass, Computing with spikes, in: Special Issue on Foundations of Information Processing, TELEMATIK 8 (1) (2002) 32–36.
[9] H. Adorna, F. Cabarle, L. Macías-Ramos, L. Pan, M.J. Pérez-Jiménez, B. Song, T. Song, L. Valencia-Cabrera, Taking the pulse of SN P systems: a quick survey, in: M. Gheorghe, I. Petre, M.J. Pérez-Jiménez, G. Rozenberg, A. Salomaa (Eds.), Multidisciplinary Creativity, Spandugino, 2015, pp. 3–16.
[10] M. Gao, M. Zhou, X. Huang, Z. Wu, Fuzzy reasoning Petri nets, IEEE Trans. Syst. Man Cybern. Syst. 33 (3) (2003) 314–324, http://dx.doi.org/10.1109/TSMCA.2002.804362.