Fuzzy reasoning spiking neural P system for fault diagnosis

Hong Peng [a,c], Jun Wang [b], Mario J. Pérez-Jiménez [c], Hao Wang [a], Jie Shao [a], Tao Wang [b]

[a] School of Mathematics and Computer Engineering, Xihua University, Chengdu 610039, China
[b] School of Electrical and Information Engineering, Xihua University, Chengdu 610039, China
[c] Research Group of Natural Computing, Department of Computer Science and Artificial Intelligence, University of Seville, Sevilla 41012, Spain

Keywords

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

A B S T R A C T

Spiking neural P systems (SN P systems) have been well established as a novel class of distributed parallel computing models. Some features that SN P systems possess are attractive to fault diagnosis. However, handling fuzzy diagnosis knowledge and reasoning is required for many fault diagnosis applications. The lack of ability is a major problem of existing SN P systems when applying them to the fault diagnosis domain. Thus, we extend SN P systems by introducing some new ingredients (such as three types of neurons, fuzzy logic and new firing mechanism) and propose the fuzzy reasoning spiking neural P systems (FRSN P systems). The FRSN P systems are particularly suitable to model fuzzy production rules in a fuzzy diagnosis knowledge base and their reasoning process. Moreover, a parallel fuzzy reasoning algorithm based on FRSN P systems is developed according to neuron's dynamic firing mechanism. Besides, a practical example of transformer fault diagnosis is used to demonstrate the feasibility and effectiveness of the proposed FRSN P systems in fault diagnosis problem.

1. Introduction

The fault diagnosis of electrical machines have moved in recent years from traditional techniques to artificial intelligence techniques [2,12,24]. In order to guarantee the secure and stable operation of electrical equipment, it is very important to diagnose its faults rapidly and accurately. More recently, a large number of methods or models using artificial intelligence and knowledge engineering have been addressed, such as expert system, Bayesian networks, artificial neural networks, and genetic algorithm [1,5,10,11,21,29]. Unfortunately, however, most of the existing methods are capable of dealing with simple diagnosis reasoning. Consequently, their reasoning processes are often difficult when complex diagnosis are shown.

Membrane computing is a class of distributed parallel computing models inspired by the structure and functioning of living cells, as well as from the way the cells are organized in tissues or higher order structures, which was introduced by Gh. Păun in 2000 [15]. In the membrane computing domain, this kind computing systems (devices) are commonly called P systems. The main ingredients of a P system are (i) *the membrane structure* (a rooted tree), delimiting compartments where (ii) *multisets of objects* evolve according to (iii) *(reaction) rules* of a bio-chemical inspiration [17]. According to their structures, these models can be divided into three categories: cell-like P systems, tissue-like P systems and neural-like P systems. Spiking neural P systems (SN P systems) firstly introduced by Ionescu et al. [8], as main forms of neural-like P systems recently investigated, are incorporated into membrane computing from the way that biological neurons communicate through electrical impulses of identical form (spikes). An SN P system can be viewed as a set of neurons placed in the nodes of a directed graph whose arcs represent the synaptic connections among the neurons. The flow of information is inherently realized by the exciting of pulse potentials, which are encoded by the so-called spikes. The spikes are objects of a unique type and are

placed inside the neurons and can be sent from presynaptic to postsynaptic neurons according to specific firing/spiking rules and by means of forgetting rules which are associated with neuron. By applying a firing/spiking rule some spikes are consumed and new spikes are produced. The spikes produced are sent to all neurons linked by a synapse to the neuron where the rule is used. By applying a forgetting rule, spikes are removed from neurons. Currently, SN P systems and a number of variants have been proposed [3,7,8,14,16,20,25,26,28]. However, only a few results applied to practical problems have been addressed [9,17,27].

This paper focuses on the application of SN P systems paradigm to the fault diagnosis problem. In addition to distributed and parallel computing advantage, the following features can be summarized from inherent mechanism of SN P systems: (i) high understandability (due to their directed graph structure), (ii) dynamically (neuron's firing and spiking mechanisms make them suitable to model dynamic behaviors of a system), (iii) synchronization (that makes them suitable to describe concurrent events or activities), (iv) non-linearly (that makes them suitable to process non-linear situation), (v) non-deterministically. As we know, fault diagnosis problem can refer to the handling of the uncertainty and incompleteness, the representation of diagnosis knowledge and reasoning [10,24]. These features and advantage of SN P systems seem to be suitable to model diagnosis knowledge and reasoning to fault diagnosis. However, major drawback of existing SN P systems and their variants is not capable of handling incomplete and uncertain information.

In this paper we propose a class of extended SN P systems, called fuzzy reasoning spiking neural P systems (FRSN P systems). Some new ingredients are introduced to extend original SN P systems, including three types of neurons (proposition neurons, "AND"-type and "OR"-type rule neurons), fuzzy truth value, fuzzy logic and new firing mechanism. The FRSN P systems can well model and visualize fuzzy production rules in a diagnosis knowledge base due to their graphical nature. Combination of neuron's new firing mechanism and fuzzy logic ensures to automatically accomplish dynamic fuzzy reasoning. Furthermore, a fuzzy reasoning algorithm based on matrix operations is developed, which can give full play to maximally parallel computing advantage of the FRSN P systems.

The present paper is organized as follows. Section 2 reviews some concepts of SN P systems, and then describes the proposed FRSN P systems. Section 3 describes a modeling method of fuzzy production rules based on the FRSN P systems. A fuzzy reasoning algorithm based on the proposed FRSN P systems is presented in Section 4. A fault diagnosis example is provided in Section 5, and conclusions are discussed in Section 6.

2. FRSN P systems

2.1. SN P systems

Here, we briefly review the basic concepts of SN P systems in the computing form (i.e., able to take some inputs and provide some outputs) (see [8,17]).

Definition 1. A SN P system of degree $m \geqslant 1$, is a construct of the form

$$\Pi = (A, \sigma_1, \ldots, \sigma_m, syn, I, O)$$

where

(1) $A = \{a\}$ is the singleton alphabet (the object a is called spike);
(2) $\sigma_1, \ldots, \sigma_m$ are neurons, of the form $\sigma_i = (n_i, r_i)$ with $i \in \{1, \ldots, m\}$, where
  (i) $n_i \geqslant 0$ is the initial number of spikes contained by neuron σ_i;
  (ii) r_i is a finite set of rules of the form

  $$E/a^c \rightarrow a^p; d$$

  where E is a regular expression over a, $c \geqslant 1$ and $p, d \geqslant 0$, with $c \geqslant p$; if $p = 0$, then E is the empty string, $d = 0$, and a^s belongs to the language generated by expression E for no rule $E/a^c \rightarrow a^p$; $d(c \geqslant p \geqslant 1)$ from r_i.
(3) $syn \subseteq \{1,2,\ldots,m\} \times \{1,2,\ldots,m\}$ with $i \neq j$ for all $(i,j) \in syn$, $1 \leqslant i, j \leqslant m$ (*synapses* between neurons).
(4) I and O are input neuron set and output neuron set, respectively.

In the above definition, the rule $E/a^c \rightarrow a^p; d$ with $p \geqslant 1$ is called *firing/spiking rule*; the rule $E/a^c \rightarrow a^p; d$ with $p = d = 0$ is written in the form $a^c \rightarrow \lambda$ and is called *forgetting rule*. The firing mechanism of neurons can be explained as follows: if a neuron σ_i contains k spikes, a^k belongs to the language generated by expression E (see [23] for further details about language theory), and $k \geqslant c$, the firing/spiking rule $E/a^c \rightarrow a^p$; $d \in r_i$ (with $p \geqslant 1$) in neuron σ_i is enabled and can be applied. This means that c spikes are consumed, $k - c$ spikes remain in the neuron, the neuron is fired, and then it produces p spikes after d time units. If $d = 0$ the spikes are emitted immediately. In the case $d \geqslant 1$, if the rule is used at step t, the neuron is "closed" or "blocked" at steps $t, t+1, \ldots, t+d-1$, and it cannot receive new spikes from other neurons. At step $t + d$, the neuron emitts spikes and becomes again open, hence it can receive spikes. The p spikes emitted by the neuron σ_i are replicated and they go to all neurons σ_j such that $(i,j) \in syn$ (each such neuron σ_j of them receives p spikes). A forgetting rule $a^c \rightarrow \lambda$ is applicable to a neuron whetheher the neuron contains exactly c spikes and then all c spikes are removed.

SN P systems are synchronized because a global clock is assumed, marking the time for the whole system. Besides, SN P systems are non-deterministic because two rules $E_1/a^{c_1} \to a^{p_1}; d_1$ and $E_2/a^{c_2} \to a^{p_2}; d_2$ can have $L(E_1) \cap L(E_2) \neq \emptyset$. Therefore, it is possible that two or more rules of the system can be enabled in a neuron. In this case, one of them is non-deterministically chosen to be used. Moreover, in each time unit, if a neuron can use a rule, the rule must be used. Each neuron deals with its spikes in the sequential manner, only using one rule in each time unit, but the rules are used in parallel for all neurons of the system.

An *instantaneous description* or a *configuration* at any instant of a SN P system is described by both the number of spikes in each neuron and the state of the neuron, more precisely, by the number of steps to count down until it becomes open (this number is zero if the neuron is already open). The *initial configuration* is described by the number of spikes initially placed in each neuron, $n_1, n_2, \ldots, n_m$, with all neurons being open. A configuration is a *halting configuration* if all neurons are open and no rule of the system is applicable to it. Using the rules described above, one can define *transitions* among configurations. We say that configuration C_1 yields configuration C_2 in one *transition step*, denoted by $C_1 \Rightarrow_{\Pi} C_2$, if we can pass from C_1 to C_2 by applying the rules from the system following the previous remarks.

A *computation* of Π is a (finite or infinite) sequence of configurations such that:

1. the first term of the sequence is the initial configuration of the system;
2. each non-initial configuration of the sequence is obtained from the previous configuration by a transtion step; and
3. if the sequence is finite (called *halting computation*) then the last term of the sequence is a halting configuration.

A computation in a system as above starts in the initial configuration. In order to compute a function f:$\mathbf{N}^k \to \mathbf{N}$ we introduce k natural numbers $q_1, \ldots, q_k$ in the system by "reading" from the environment a binary sequence $z = 10^{q_1-1}10^{q_2-1}1 \ldots 10^{q_k-1}1$ This means that the input neuron of Π receives a spike in each step corresponding to a digit 1 from the string z and no spike otherwise. Note that we input exactly $k+1$ spikes, i.e., after the last spike we assume that no further spike is coming to the input neuron.

With any computation (halting or not) we can associate a *spike train*, sequence of symbols 0 and 1 describing the behavior of the output neuron: if the output neuron spikes, then we write 1; otherwise we write 0. Besides, we also associate other forms of computation results according to different computing purposes, such as, the distance between two consecutive steps when there are spikes which exit the system.

2.2. FRSN P systems

The goal of this paper is to develop a new modeling method for fault diagnosis inspired by SN P systems. As stated above, fault diagnosis can involve such a problem, that is, representation of fuzzy diagnosis knowledge and fuzzy reasoning. However, current forms of existing SN P systems and their variants cannot deal with the problem. In order to make them able to handle fuzzy diagnosis knowledge and fuzzy reasoning, we will extend the definition of SN P systems and propose a new class of extended SN P systems, called *fuzzy reasoning spiking neural* P *systems* (FRSN P systems, in short).

Definition 2. A FRSN P system of degree $m \geqslant 1$, is a construct of the form

$$\Pi = (A, \sigma_1, \ldots, \sigma_m, syn, I, O)$$

where

(1) $A = \{a\}$ is the singleton alphabet (the object a is called spike);
(2) $\sigma_1, \ldots, \sigma_m$ are neurons, of the form $\sigma_i = (\alpha_i, \tau_i, r_i)$ with $i \in \{1, \ldots, m\}$ where
  (i) $\alpha_i \in [0,1]$ and it is called the (potential) value of spike contained in neuron σ_i (also called pulse value);
  (ii) $\tau_i \in [0,1]$ is the truth value associated with neuron σ_i;
  (iii) r_i is a firing/spiking rule contained in neuron σ_i, of the form $E/a^{\alpha} \to a^{\beta}$, where $\alpha, \beta \in [0,1]$.
(3) $syn \subseteq \{1,2,\ldots,m\} \times \{1,2,\ldots,m\}$ with $i \neq j$ for all $(i,j) \in syn$, $1 \leqslant i, j \leqslant m$ (*synapses* between neurons);
(4) I and O are input neuron set and output neuron set, respectively.

Now, we explain how the FRSN P systems are extended from SN P systems. First of all, content of neuron is denoted by a fuzzy truth value in $[0,1]$ instead of the number of spikes in SN P systems, which can be interpreted as the (potential) value of spike from the view point of biological neuron. For a neuron σ_i, if $\alpha_i > 0$, we say the neuron contains a spike with (potential) value α_i; otherwise, the neuron contains no spike. Secondly, each neuron in FRSN P systems will associate with either a fuzzy proposition or a fuzzy production rule, and $\tau_i \in [0,1]$ will be used to express the truth value of the fuzzy proposition or confidence factor (CF) of the fuzzy production rule. Thirdly, each neuron contains only one spiking (firing) rule, of the form $E/a^{\alpha} \to a^{\beta}$, where $E = a^n$ is called the firing condition and n is the number of input synapses from other neurons to the neuron. The firing condition $E = a^n$ indicates that if the neuron receives n spikes the spiking rule can be applied; otherwise the rule cannot be enabled until n spikes are received. When the number of spikes received by a neuron is less than n, value of the spikes received will be updated according to logical "AND" or "OR" operations. Fourthly, the firing mechanism of neurons in FRSN P systems can be described as follows. For the neuron σ_i, if its firing rule $E/a^{\alpha} \to a^{\beta}$ can be applied, this means that its

pulse value $\alpha > 0$ is consumed (removed), the neuron fires, and then it produces a spike with value β. Once the spike with value β is excited from neuron σ_i, all neurons σ_j with $(i,j) \in syn$ will immediately receive the spike. Further, three types of neurons are defined (see Definitions 3–5 below). The three types of neurons use different ways to handle both α and β, and β is relative with both α and τ_i. Finally, time delay is ignored in FRSN P system, thus all neurons are always open.

As stated above, neurons in FRSN P system are classified into three classes: proposition neuron, "AND"-type rule neuron and "OR"-type rule neuron. These types of neurons are defined as follows.

Definition 3. The proposition neurons are a class of neurons, which are associated with propositions in a fuzzy knowledge base.

A proposition neuron can be denoted by $\sigma = (\alpha, \tau, r)$, where α is its pulse value, τ is the truth value of the proposition associated with it, and r is its spiking rule of the form $E/a^{\alpha} \to a^{\alpha}$. If a proposition neuron σ is an input neuron in Π, then we have $\alpha = \tau$; otherwise, α equals to logical "OR" operation of all pulse values received from other neurons. After the neuron updates its content, the truth value of the corresponding proposition will be equal to its pulse value, i.e., $\tau = \alpha$. When the neuron fires and applies its firing rule, it will produce a spike with pulse value α. Fig. 1 shows a proposition neuron.

Definition 4. The "AND"-type rule neurons are a class of neurons, which are associated with fuzzy production rules with "AND"-type antecedent part, where confidence factor (CF) of each the rule is denoted by τ.

Fig. 2 shows a "AND"-type rule neuron, which is labeled by the symbol "AND". When a "AND"-type rule neuron receives n pulse values from other neurons, $\alpha_1, \alpha_2, \ldots, \alpha_n$, it uses logic operator "AND" to combine its all inputs, i.e., $\alpha = \min(\alpha_1, \alpha_2, \ldots, \alpha_n)$, where $\alpha_i \in [0, 1]$, $1 \leqslant i \leqslant n$. If its firing condition E is satisfied, it fires and applies its spiking rule $E/a^{\alpha} \to a^{\beta}$ to produce a spike with value $\beta = \alpha * \tau$, hence $\beta = \min(\alpha_1, \alpha_2, \ldots, \alpha_n) * \tau$; otherwise, it only updates its content by using pulse values of the received pikes.

Definition 5. The "OR"-type rule neurons are a class of neurons, which are associated with fuzzy production rules with "OR"-type antecedent part, where confidence factor (CF) of each the rule is denoted by τ.

Fig. 3 shows a "OR"-type rule neuron, which is labeled by the symbol "OR". When a "OR"-type rule neuron receives n pulse values from other neurons, $\alpha_1, \alpha_2, \ldots, \alpha_n$, it uses the logic operator "OR" to combine its all inputs, i.e., $\alpha = \max(\alpha_1, \alpha_2, \ldots, \alpha_n)$, where $\alpha_i \in [0, 1]$, $1 \leqslant i \leqslant n$. Similarly, when E is satisfied, it fires and applies its spiking rule $E/a^{\alpha} \to a^{\beta}$ to produce a spike with value $\beta = \alpha * \tau$, hence $\beta = \max(\alpha_1, \alpha_2, \ldots, \alpha_n) * \tau$; otherwise, it only updates its content by using pulse values of the received pikes.

3. Modeling fuzzy production rules using FRSN P systems

3.1. Fuzzy production rules

In fault diagnosis application, diagnosis knowledge extracted from real-world data are usually expressed by fuzzy production rules. To date, fuzzy production rule is a widely used tool for fuzzy knowledge representation (see [4,6,13,19,18,22,30]). Usually, a simple fuzzy production rule is of the form

$$R_i: \text{ IF } p_j \text{ THEN } p_k(\text{CF} = \tau_i) \tag{1}$$

where R_i indicates ith fuzzy production rule of a fuzzy diagnosis knowledge base and $\tau_i \in [0, 1]$ is its confidence factor. p_j and p_k are two propositions and their truth values are real numbers in $[0, 1]$.

In addition to above simple fuzzy production rule, composite fuzzy production rules are widely used in fuzzy diagnosis knowledge base, where their antecedent part or consequence part usually contain "and" or "or" connectors. Generally, composite fuzzy production rules can be classified as the following three types.

$$\text{Type 1 } R_i: \text{ IF } p_1 \quad \text{and} \quad p_2 \quad \text{and} \ldots \text{and } p_{k-1} \text{ THEN } p_k \ (\text{CF} = \tau_i) \tag{2}$$

The type rule is a composite conjunctive fuzzy production rule, where $p_1, p_2, \ldots, p_{k-1}$ are propositions in the antecedent part of the rule, and $\tau_i \in [0, 1]$ is a real number and it expresses the confidence factor of the rule. Suppose truth values of propositions $p_1, p_2, \ldots, p_{k-1}$ are $\alpha_1, \alpha_2, \ldots, \alpha_{k-1}$ respectively. Then, truth value of proposition p_k can be evaluated as $\alpha_k = \min(\alpha_1, \alpha_2, \ldots, \alpha_{k-1}) * \tau_i$.

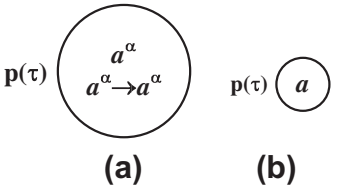

Fig. 1. (a) A proposition neuron P and (b) its simple form.

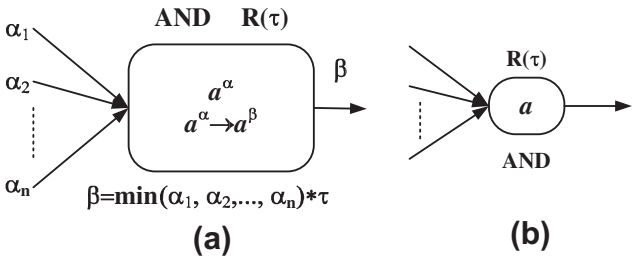

Fig. 2. (a) A "AND"-type rule neuron and (b) its simple form.

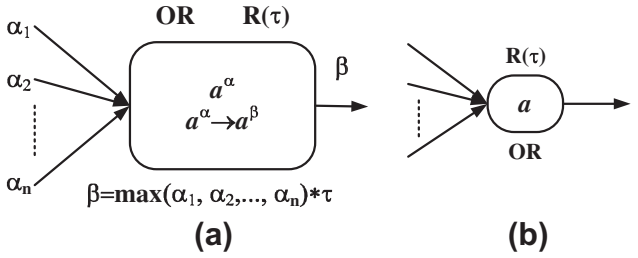

Fig. 3. (a) A "OR"-type rule neuron and (b) its simple form.

$$\text{Type 2 } R_i:\ \text{IF } p_1 \text{ THEN } p_2 \text{ and } p_3 \text{ and } \cdots \text{ and } p_k\ (\text{CF} = \tau_i) \tag{3}$$

where $\tau_i \in [0,1]$ is the confidence factor of the rule and p_1 is alone proposition in the antecedent part of the rule. Suppose the truth value of proposition p_1 is α_1. Then, truth values of propositions $p_2, p_3, \ldots, p_k$ can be evaluated as $\alpha_2 = \alpha_1 * \tau_i$, $\alpha_3 = \alpha_1 * \tau_i, \ldots, \alpha_k = \alpha_1 * \tau_i$, respectively.

$$\text{Type 3 } R_i:\ \text{IF } p_1 \text{ or } p_2 \text{ or } \cdots \text{ or } p_{k-1} \text{ THEN } p_k\ (\text{CF} = \tau_i) \tag{4}$$

The type rule is a composite disjunctive fuzzy production rule, where $p_1, p_2, \ldots, p_{k-1}$ are propositions in antecedent part of the rule, and $\tau_i \in [0,1]$ is the confidence factor of the rule. Suppose truth values of propositions $p_1, p_2, \ldots, p_{k-1}$ are $\alpha_1, \alpha_2, \ldots, \alpha_{k-1}$, respectively. Then, truth value of proposition p_k can be evaluated as $\alpha_k = \max(\alpha_1, \alpha_2, \ldots, \alpha_{k-1}) * \tau_i$.

3.2. FRSN P system model for fuzzy production rules

In order to model fuzzy production rules of a fuzzy diagnosis knowledge base, we need map them into a FRSN P system model. The basic principle can be described as follows. Each proposition in the fuzzy diagnosis knowledge base is represented by a proposition neuron, while each fuzzy production rule is expressed by a rule neuron ("AND"-type neuron or "OR"-type neuron). At the initial stage, each input proposition neuron contains a spike and its pulse value is assigned to the truth value of the proposition associated with it. Moreover, value τ_i of each rule neuron is assigned to the confidence factor of the fuzzy production rule associated with it. Consequently, fuzzy production rules of a fuzzy diagnosis knowledge base can be modeled by FRSN P systems, and their dynamic reasoning process can be realized by the firing mechanism of neurons.

According to the above principle, simple fuzzy production rule (1) can be modeled by the following FRSN P system Π_0. Fig. 4a shows the model Π_0 and its reasoning result.

$\Pi_0 = (A, \sigma_i, \sigma_j, \sigma_k, syn, I, O)$, where

(1) $A = \{a\}$
(2) σ_i is a rule neuron associated with rule R_i with confidence factor τ_i. Its spiking rule is of the form $E/a^{\alpha} \to a^{\beta}$, where $\beta = \alpha * \tau_i$.
(3) σ_j and σ_k are two proposition neurons associated with propositions p_j and p_k with truth values α_j and α_k respectively. Their spiking rules are of the form $E/a^{\alpha} \to a^{\alpha}$.
(4) $syn = \{(j,i), (i,k)\}$, $I = \{\sigma_j\}$, $O = \{\sigma_k\}$.

A type-1 composite fuzzy production rule (2) can be modeled by the following FRSN P system Π_1. Fig. 4b shows the model Π_1 and its reasoning result.

$\Pi_1 = (A, \sigma_1, \sigma_2, \ldots, \sigma_k, \sigma_{k+1}, syn, I, O)$, where

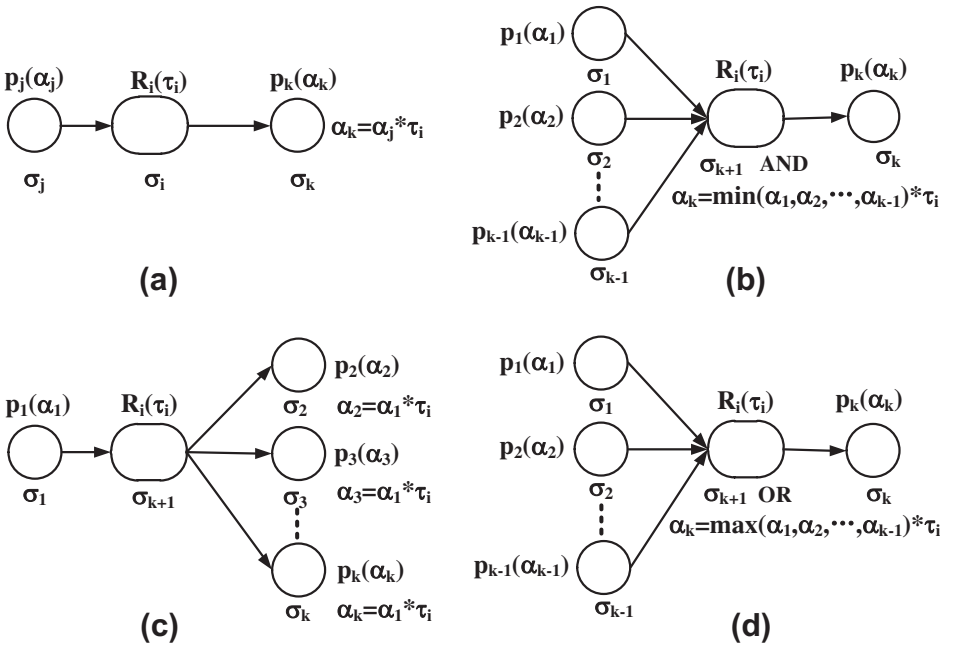

Fig. 4. Fuzzy production rules modeled by ERSN P systems and their rule reasoning: (a) simple rule; (b) type-1 composite rule; (c) type-2 composite rule; (d) type-3 composite rule.

(1) $A = \{a\}$
(2) $\sigma_j (j = 1, 2, \ldots, k)$ are proposition neurons associated with propositions $p_j (j = 1, 2, \ldots, k)$ with truth values $\alpha_j (j = 1, 2, \ldots, k)$ respectively. Their spiking rules are of the form $E/a^{\alpha} \rightarrow a^{\alpha}$.
(3) σ_{k+1} is a "AND"-type rule neuron associated with rule R_i with confidence factor τ_i. Its spiking rule is of the form $E/a^{\alpha} \rightarrow a^{\beta}$, where $\beta = \alpha * \tau_i$.
(4) $syn = \{(1, k+1), (2, k+1), \ldots, (k-1, k+1), (k+1, k)\}$.
(5) $I = \{\sigma_1, \sigma_2, \ldots, \sigma_{k-1}\}$, $O = \{\sigma_k\}$.

A type-2 composite fuzzy production rule (3) can be modeled by the following FRSN P system Π_2. Fig. 4c shows the model Π_0 and its reasoning result.

$\Pi_2 = (A, \sigma_1, \sigma_2, \ldots, \sigma_k, \sigma_{k+1}, syn, I, O)$, where

(1) $A = \{a\}$
(2) σ_j $(j = 1, 2, \ldots, k)$ are proposition neurons associated with propositions p_j $(j = 1, 2, \ldots, k)$ with truth values α_j $(j = 1, 2, \ldots, k)$ respectively. Their spiking rules are of the form $E/a^{\alpha} \rightarrow a^{\alpha}$.
(3) σ_{k+1} is a rule neuron associated with rule R_i with confidence factor τ_i. Its spiking rule is of the form $E/a^{\alpha} \rightarrow a^{\beta}$, where $\beta = \alpha * \tau_i$.
(4) $syn = \{(1, k+1), (k+1,2), (k+1,3), \ldots, (k+1, k)\}$.
(5) $I = \{\sigma_1\}$, $O = \{\sigma_2, \sigma_3, \ldots, \sigma_k\}$.

A type-3 composite fuzzy production rule (4) can be modeled by the following FRSN P system Π_3. Fig. 4d shows the model Π_0 and its reasoning result.

$\Pi_3 = (A, \sigma_1, \sigma_2, \ldots, \sigma_k, \sigma_{k+1}, syn, I, O)$, where

(1) $A = \{a\}$
(2) σ_j $(j = 1, 2, \ldots, k)$ are proposition neurons associated with propositions p_j $(j = 1, 2, \ldots, k)$ with truth values α_j $(j = 1, 2, \ldots, k)$ respectively. Their spiking rules are of the form $E/a^{\alpha} \rightarrow a^{\alpha}$.
(3) σ_{k+1} is an "OR"-type rule neuron associated with rule R_i with confidence factor τ_i. Its spiking rule is of the form $E/a^{\alpha} \rightarrow a^{\beta}$, where $\beta = \alpha * \tau_i$.
(4) $syn = \{(1, k+1), (2, k+1), \ldots, (k-1, k+1), (k+1, k)\}$.
(5) $I = \{\sigma_1, \sigma_2, \ldots, \sigma_{k-1}\}$, $O = \{\sigma_k\}$.

4. Reasoning algorithm based on FRSN P systems

In this section, we will present a fuzzy reasoning algorithm based on FRSN P systems. Suppose all fuzzy production rules in a fuzzy diagnosis knowledge base have been modeled by a FRSN P system model Π. The model Π consists of m neurons, where $\sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ are its n proposition neurons, $\bar{\sigma} = \{\bar{\sigma}_1, \bar{\sigma}_2, \ldots, \bar{\sigma}_k\}$ are its k rule neurons ("AND" type neurons or 'OR" type neurons), $m = n + k$. Usually, we should provide the fuzzy truth values for a part of the fuzzy propositions before reasoning, and proposition neurons associated with the part of fuzzy propositions are indeed input neurons of Π. The goal of the reasoning algorithm is to reason out the fuzzy truth values of other unknown fuzzy propositions (proposition neurons) from known fuzzy propositions (input neurons). These unknown fuzzy propositions are often associated with output neurons of Π. The reasoning algorithm developed here is based on neuron's firing mechanism and uses matrix operation. As a result, The reasoning algorithm can give full play to the advantages of parallel computing of FRSN P systems.

In the following, in order to succinctly illustrate the fuzzy reasoning algorithm, we introduce some matrix and vector notations as follows.

(1) $U = (u_{ij})_{n\times k}$ is a binary matrix, where $u_{ij} \in \{0,1\}$. $u_{ij} = 1$ if there is a directed arc (synapse) from proposition neuron σ_i to rule neuron $\bar{\sigma}_j$. $u_{ij} = 0$ if there is no directed arc (synapse) from proposition neuron σ_i to rule neuron $\bar{\sigma}_j$.
(2) $V = (v_{ij})_{n\times k}$ is a binary matrix, where $v_{ij} \in \{0,1\}$. $v_{ij} = 1$ if there is a directed arc (synapse) from rule neuron $\bar{\sigma}_j$ to proposition neuron σ_i. $v_{ij} = 0$ if there is no directed arc (synapse) from rule neuron $\bar{\sigma}_j$ to proposition neuron σ_i.
(3) $\Lambda = \mathrm{diag}(\tau_1, \tau_2, \ldots, \tau_k)$ is a diagonal matrix, where τ_i represents confidence factor of ith production rule associated with rule neuron $\bar{\sigma}_i, 1 \leqslant i \leqslant k$.
(4) $H_1 = \mathrm{diad}(h_1, h_2, \ldots, h_k)$ is a diagonal matrix. If the ith rule neuron is an "AND"-type neuron, $h_i = 1$; otherwise $h_i = 0$.
(5) $H_2 = \mathrm{diad}(h_1, h_2, \ldots, h_k)$ is a diagonal matrix. If the ith rule neuron is an "OR"-type neuron, $h_i = 1$; otherwise $h_i = 0$.
(6) $\alpha_p = (\alpha_{p1}, \alpha_{p2}, \ldots, \alpha_{pn})^T$ is a truth value vector, $\alpha_{pi} \in [0,1]$. α_{pi} represents the truth value of ith proposition neuron. $\alpha_r = (\alpha_{r1}, \alpha_{r2}, \ldots, \alpha_{rk})^T$ is also a truth value vector, $\alpha_{rj} \in [0,1]$. α_{rj} represents the truth value of jth rule neuron.
(7) $a_p = (a_{p1}, a_{p2}, \ldots, a_{pn})^T$ is an integer vector, where a_{pi} represents the number of spikes received by ith proposition neuron. $a_r = (a_{r1}, a_{r2}, \ldots, a_{rk})^T$ is also an integer vector, where a_{rj} represents the number of spikes received by jth rule neuron.
(8) $\lambda_p = (\lambda_{p1}, \lambda_{p2}, \ldots, \lambda_{pn})^T$ is an integer vector, where λ_{pi} represents the number of spikes required by firing ith proposition neuron. $\lambda_r = (\lambda_{r1}, \lambda_{r2}, \ldots, \lambda_{rk})^T$ is also an integer vector, where λ_{rj} represents the number of spikes required by firing jth rule neuron.
(9) $\beta_p = (\beta_{p1}, \beta_{p2}, \ldots, \beta_{pn})^T$ is a truth value vector, $\beta_{pi} \in [0,1]$. β_{pi} represents truth value exported by ith proposition neuron after firing. $\beta_r = (\beta_{r1}, \beta_{r2}, \ldots, \beta_{rk})^T$ is also a truth value vector, $\beta_{rj} \in [0,1]$. β_{rj} represents the truth value exported by jth rule neuron after firing.
(10) $b_p = (b_{p1}, b_{p2}, \ldots, b_{pn})^T$ is an integer vector, where $b_{pi} \in \{0,1\}$ represents the number of spikes exported by ith proposition neuron after firing. $b_r = (b_{r1}, b_{r2}, \ldots, b_{rk})^T$ is also an integer vector, where $b_{rj} \in \{0,1\}$ represents the number of spikes exported by jth rule neuron after firing.

In addition to above notations, we also introduce the following several operators and functions.

(1) $\oplus$: $C = A \oplus B$, where A, B and C are all $r \times s$ matrices, such that $c_{ij} = \max\{a_{ij}, b_{ij}\}$.
(2) $\otimes$: $C = A \otimes B$, where A, B and C are $r \times s$, $s \times t$ and $r \times t$ matrices respectively, such that $c_{ij} = \max_{1\leqslant r\leqslant s}\{a_{ir} \cdot b_{rj}\}$.
(3) $\odot$: $C = A \odot B$, where A, B and C are $r \times s$, $s \times t$ and $r \times t$ matrices respectively, such that $c_{ij} = \min_{1\leqslant r\leqslant s}\{a_{ir} \cdot b_{rj}\}$.
(4) $\beta = \mathit{fire}(\alpha, a, \lambda)$, where $\beta = (\beta_1, \ldots, \beta_r)^T$, $\alpha = (\alpha_1, \ldots, \alpha_r)^T$, $a = (a_1, \ldots, a_r)^T$, $\lambda = (\lambda_1, \ldots, \lambda_r)^T$. The function is defined as follows:

$$\beta_i = \begin{cases} \alpha_i, & \text{if } a_i = \lambda_i \\ 0 & \text{if } a_i < \lambda_i \end{cases},$$

where $i = 1, 2, \ldots, r$.
(5) $\beta = \mathit{update}(\alpha, a, \lambda)$, where β, α, a and λ are vectors described above. The function is defined as follows:

$$\beta_i = \begin{cases} 0 & \text{if } a_i = 0 \\ \beta_i + \alpha_i, & \text{if } 0 < a_i < \lambda_i \\ 0 & \text{if } a_i = \lambda_i \end{cases},$$

where $i = 1, 2, \ldots, r$.
(6) D=$\mathit{diag}(b)$, where $D = (d_{ij})$ is a $r \times r$ diagonal matrix and $b = (b_1, \ldots, b_r)$. For $1 \leqslant i \leqslant r$, $d_{ii} = b_i$, while $d_{ij} = 0$ for $i \neq j$.

Then, the developed fuzzy reasoning algorithm based on FRSN P system is described as follows.

Fuzzy reasoning algorithm based on FRSN P system.

INPUT: parameter matrixes U, V, Λ, H_1, H_2, λ_p, λ_r, and initial inputs α_p^0, a_p^0.

OUTPUT: The fuzzy truth values of propositions associated with the neurons in O.

Step 1) Let $\alpha_r^0 = (0,0,\ldots,0)^T, a_r^0 = (0,0,\ldots,0)^T$.

Step 2) Let $t = 0$.

Step 3)

(1) Process the firing of proposition neurons.

$$\beta_p^t = fire\left(\alpha_p^t, a_p^t, \lambda_p\right), b_p^t = fire\left(1, a_p^t, \lambda_p\right), \alpha_p^t = update\left(\alpha_p^t, a_p^t, \lambda_p\right),$$
$$a_p^t = update\left(a_p^t, a_p^t, \lambda_p\right), B_p^t = diag\left(b_p^t\right).$$

(2) Compute the truth values of rule neurons and the number of received spikes.

$$\alpha_r^{t+1} = \alpha_r^t \oplus \left[\left(H_1 \cdot \left((B_p^t \cdot U)^T \odot \beta_p^t\right)\right) + (H_2 \cdot \left(\left(B_p^t \cdot U\right)^T \otimes \beta_p^t\right)\right],$$
$$a_r^{t+1} = a_r^t + \left[\left(B_p^t \cdot U\right)^T \cdot b_p^t\right].$$

(3) Process the firing of rule neurons.

$$\beta_r^{t+1} = fire(\Lambda \cdot \alpha_r^{t+1}, a_r^{t+1}, \lambda_r), b_r^{t+1} = fire(1, a_r^{t+1}, \lambda_r),$$
$$\alpha_r^{t+1} = update\left(\alpha_r^{t+1}, a_p^t, \lambda_p\right), a_r^{t+1} = update\left(a_r^{t+1}, a_p^t, \lambda_p\right), B_r^{t+1} = diag\left(b_r^{t+1}\right).$$

(4) Compute the truth values of proposition neurons and the number of received spikes.

$$\alpha_p^{t+1} = \alpha_p^t \oplus \left[\left(V \cdot B_r^{t+1}\right) \otimes \beta_r^{t+1}\right], a_p^{t+1} = a_p^t + \left[\left(V \cdot B_r^{t+1}\right) \cdot b_r^{t+1}\right].$$

Step 4) If $a_p^{t+1} = (0,0,\ldots,0)^T$ and $a_r^{t+1} = (0,0,\ldots,0)^T$ (computation halts), the reasoning results are obtained; otherwise, $t = t + 1$, go to Step 3).

5. Application to fault diagnosis

As discussed above, fuzzy production rules can be modeled by the proposed FRSN P system and the developed reasoning algorithm is a parallel fuzzy reasoning algorithm. In this section, a practical example is used to illustrate application of the proposed method to a transformer fault diagnosis system. The following fuzzy production rules are from knowledge base of a transformer fault diagnosis system.

Rule 1 (*Confidence* = 0.8)
Symptom:
(1) Total hydrocarbon is little high (p_1).
(2) C_2H_2 is low (p_2).
Anticipated Fault: General overheating fault occurs (p_{11}).
Rule 2 (*Confidence* = 0.8)
(1) Total hydrocarbon is rather high (p_3).
(2) C_2H_2 is too high (p_4).
(3) H_2 is high (p_5).
(4) C_2H_2 in total hydrocarbon occupies a too low proportion p_6
Anticipated Fault: Serious overheating fault occurs (p_{11}).
Rule 3 (*Confidence* = 0.8)
(1) Total hydrocarbon is little low (p_7).
(2) H_2 is high (p_5).
(3) CH_4 in total hydrocarbon occupies a large proportion p_8
(4) CH_4 in total hydrocarbon occupies a higher proportion than C_2H_2 p_9
Anticipated Fault: The partial discharge occurs (p_{13}).
Rule 4 (*Confidence* = 0.8)
(1) Total hydrocarbon is rather low (p_{10}).
(2) C_2H_2 is too high (p_4).

(3) H_2 is high (p_5).
Anticipated Fault: The spark discharge occurs (p_{14}).

These fuzzy production rules can be modeled by the following FRSN P system Π_4, shown in Fig. 5.
$\Pi_4 = (A, \sigma_1, \ldots, \sigma_{14}, \sigma_{15}, \ldots, \sigma_{18}, syn, I, O)$, where

(1) $A = \{a\}$.
(2) $\sigma_1, \ldots, \sigma_{14}$ are proposition neurons associated with propositions $p_1, \ldots, p_{14}$ respectively.
(3) $\sigma_{15}, \ldots, \sigma_{18}$ are "AND"-type rule neurons associated with production rules $R_1, \ldots, R_4$ respectively.
(4) $syn = \{(1,15), (2,15), (3,16), (4,16), (4,18), (5,16), (5,17), (5,18), (6,18), (7,17), (8,17), (9,17), (10,18), (15,11), (16,12), (17,13), (18,14)\}$.
(5) $I = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7, \sigma_8, \sigma_9, \sigma_{10}\}$, $O = \{\sigma_{11}, \sigma_{12}, \sigma_{13}, \sigma_{14}\}$.

According to the definition of matrices and vectors given above, U, V, Λ, H_1 and H_2 are follows:

$$U = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad H_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$V = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \quad H_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} 0.8 & 0 & 0 & 0 \\ 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0.8 \end{bmatrix}$$

$$\lambda_p = (1,1,1,1,1,1,1,1,1,1,1,1,1,1)^T \qquad \lambda_r = (2,4,4,3)^T$$

In on-scene information detection of transformer, total hydrocarbon content is high (confidence = 0.8), C_2H_2 content is high (confidence = 0.8), H_2 content is high (confidence = 0.9), C_2H_2 content in total hydrocarbon content is little (confidence = 0.8), CH_4 content in total hydrocarbon content is little (confidence = 0.1). Thus, initial truth value vector $\alpha_p^0 = (0.8, 0.2, 0.8, 0.8, 0.9, 0.8, 0.2, 0.9, 0.1, 0.2, 0, 0, 0, 0)^T$ and initial spike vector $a_p^0 = (1,1,1,1,1,1,1,1,1,1,0,0,0,0)^T$. Let $\alpha_r^0 = (0,0,0,0)^T$ and $a_r^0 = (0,0,0,0)^T$.

According to reasoning algorithm described above, we have

(1) $\alpha_p^1 = (0,0,0,0,0,0,0,0,0,0,0,0,0,0)^T$, $\alpha_r^1 = (0.16, 0.64, 0.08, 0.16)^T$,
$a_p^1 = (0,0,0,0,0,0,0,0,0,0,0,0,0,0)^T$, $a_r^1 = (2,4,4,3)^T$.

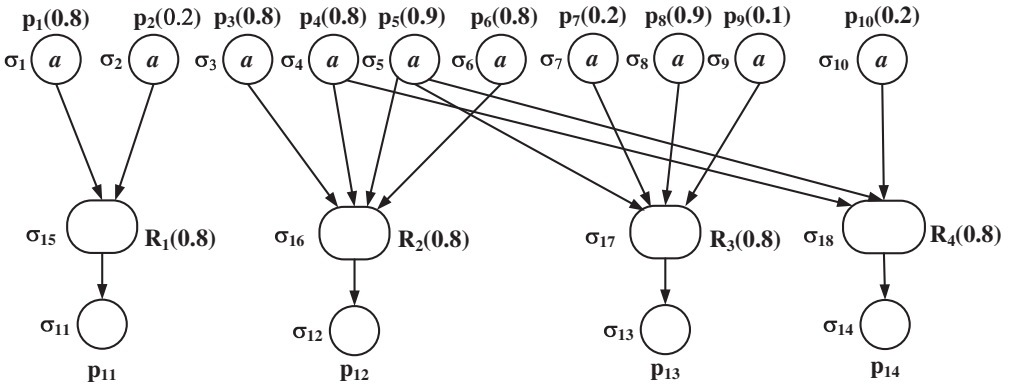

Fig. 5. An example of a transformer fault diagnosis modeled by the FRSN P system model Π_4.

(2) $\alpha_p^2 = (0,0,0,0,0,0,0,0,0,0,0.16,0.64,0.08,0.16)^T$, $\alpha_r^2 = (0,0,0,0)^T$,
$a_p^2 = (0,0,0,0,0,0,0,0,0,0,1,1,1,1)^T$, $a_r^2 = (0,0,0,0)^T$.
(3) $a_p^3 = (0,0,0,0,0,0,0,0,0,0,0,0,0,0)^T$, $a_r^3 = (0,0,0,0)^T$.

Thus, since the system reaches halting computation ($a_p^3 = (0,0,0,0,0,0,0,0,0,0,0,0,0,0)^T$ and $a_r^3 = (0,0,0,0)^T$), system exports its reasoning results, i.e., the truth values of propositions p_{11}, p_{12}, p_{13} and p_{14} are 0.16, 0.64, 0.08 and 0.16 respectively. These reasoning results indicate the following possible faults: general overheating fault (confidence = 0.16), serious overheating fault (confidence = 0.64), partial discharge (confidence = 0.08) and spark discharge (confidence = 0.16). In the fault diagnosis system, the threshold value of fault occurrence is set to be 0.6. Thus, we can conclude that the transformer shows a serious overheating fault, which is consistent with the actual situation.

6. Conclusions

In this paper, we present a novel model for fault diagnosis, called fuzzy reasoning spiking neural P systems (FRSN P systems). The FRSN P system model can visually represent fuzzy production rules of a fuzzy diagnosis knowledge base and effectively model the corresponding dynamic reasoning behavior because of its nature of directed graph and firing mechanism of neurons. Based on parallel computing feature of FRSN P systems and dynamic firing mechanism of neurons, a parallel fuzzy reasoning algorithm is developed. Moreover, a practical example of transformer fault diagnosis is used to demonstrate the feasibility and effectiveness of FRSN P systems in the fault diagnosis area. From another point of view, FRSN P systems are new extended forms of SN P systems, which can handle fuzzy information and express fuzzy knowledge. This will open a door for more practical applications that need to deal with fuzzy information.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (Grant No. 61170030), Research Fund of Sichuan Key Laboratory of Intelligent Network Information Processing (No. SGXZD1002-10), and the Importance Project Foundation of Xihua University (No. Z1122632), China. The work of Mario J. Pérez-Jiménez was supported by the project TIN2009–13192 of the Ministerio de Educación y Ciencia of Spain, cofinanced by FEDER funds, and the support of the Project of Excellence with *Investigador de Reconocida Valía* of the Junta de Andalucía, Grant P08-TIC-04200.

References

[1] M. Barakat, F. Druaux, D. Lefebvre, M. Khalil, O. Mustapha, Self adaptive growing neural network classifier for faults detection and diagnosis, Neurocomputing 74 (18) (2011) 3865–3876.
[2] P. Bhowal, D. Sarkar, S. Mukhopadhyay, A. Basu, Fault diagnosis in discrete time hybrid systems – a case study, Information Sciences 177 (5) (2007) 1290–1308.
[3] M. Cavaliere, O.H. Ibarra, Gh. Păun, O. Egecioglu, M. Ionescu, S. Woodworth, Asynchronous spiking neural P systems, Theoretical Computer Science 410 (24-25) (2009) 2352–2364.
[4] T.-C. Chiang, C.-F. Tai, T.-W. Hou, A knowledge-based inference multicast protocol using adaptive fuzzy Petri nets, Expert Systems with Applications 36 (4) (2009) 8115–8123.
[5] S. Deng, Seng-Yi Lin, We-Luan Chang, Application of multiclass support vector machines for fault diagnosis of field air defense gun, Expert Systems with Applications 38 (5) (2011) 6007–6013.
[6] M. Gao, M. Zhou, Y. Tang, Intelligent deision making in disassembly process based on fuzzy reasoning petri nets, IEEE Transaction Systems, Man, Cybernetics – Part B: Cybernetics 34 (5) (2004) 2029–2034.
[7] L. Huang, I.H. Suh, A. Abraham, Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants, Information Sciences 181 (11) (2011) 2370–2391.
[8] M. Ionescu, Gh. Păun, T. Yokomori, Spiking neural P systems, Fundameta Informaticae 71 (2-3) (2006) 279–308.
[9] M. Ionescu, D. Sburlan, Some applications of spiking neural P systems, in: Proceedings of the Eighth Workshop on Membrane Computing, 2007, pp. 383–394.
[10] S.M. Islam, T. Wu, G. Ledwich, A novel fuzzy logic approach to transformer fault diagnosis, IEEE Transactions an Dielectrics and Elecfrical Insulation 7 (2) (2000) 177–186.
[11] C.H. Lo, Y.K. Wong, A.B. Rad, Bond graph based Bayesian network for fault diagnosis, Applied Soft Computing 11 (1) (2011) 1208–1212.
[12] C. Ma, X. Gu, Y. Wang, Fault diagnosis of power electronic system based on fault gradation and neural network group, Neurocomputing 72 (13-15) (2009) 2909–2914.
[13] V. Novak, Fuzzy Set and Their Application, Adam Hilger, Bristol, 1989.
[14] L.Q. Pan, Gh. Păun, Spiking neural P systems: an improved normal form, Theoretical Computer Science 411 (6) (2010) 906–918.
[15] Gh. Păun, Computing with membranes, Journal of Computer System Sciences 61 (1) (2000) 108–143 (and TUCS Research Report 208, 1998).
[16] Gh. Păun, M.J. Pérez-Jiménez, G. Rozenberg, Spike train in spiking neural P systems, International Journal of Foundations of Computer Science 17 (4) (2006) 975–1002.
[17] Gh. Păun, G. Rozenberg, A. Salomaa, The Oxford Handbook of Membrance Computing, Oxford Unversity Press, New York, 2010.
[18] W. Pedrycz, Geeralized fuzzy petri nets as pattern classifiers, Pattern Recognition Letters 20 (1999) 769–778.
[19] W. Pedrycz, F. Gomide, A generalized fuzzy petri net model, IEEE Transaction on Fuzzy Systems 2 (1994) 295–301.
[20] H. Peng, J. Wang, Adaptive spiking neural P systems, in: 2010 Sixth International Conference of Natural Computing (ICNC2010), vol. 6, 2010, pp. 3008–3011.
[21] G.G. Rigatos, Fault detection and isolation based on fuzzy automata, Information Sciences 179 (12) (2009) 1893–1902.
[22] S. Ribaric, T. Hrkac, A model of fuzzy spatio-temporal knowledge representation and reasoning based on high-level Petri nets, Information Systems 37 (3) (2012) 238–256.
[23] G. Rozenberg, A. Salomaa, Handbook of Formal Languages, Word, Language, Grammar, vol. 1, Springer Varlag, Berlin, 1997.

[24] J. Sun, S.-Y. Qin, Y.-H. Song, Fault diagnosis of electric power systems based on fuzzy petri nets, IEEE Transaction on Power Systems 19 (4) (2004) 2053–2059.
[25] J. Wang, H.J. Hoogeboom, L. Pan, Gh. Păun, Spiking neural P systems with weights and thresholds, in: Tenth Workshop on Membrane Computing (WMC10), August, Romania, 2009, pp. 514–533.
[26] J. Wang, H. Peng, Fuzzy knowledge representation based on an improving spiking neural P system, in: 2010 Sixth International Conference of Natural Computing (ICNC2010), vol. 6, 2010, pp. 3012–3015.
[27] T. Wang, H. Peng, J. Shao, A thresholding method based on P systems for image segmentation, ICIC Express Letters 6 (1) (2012) 221–227.
[28] J. Wang, L. Zhou, H. Peng, G. Zhang, An extended spiking neural P system for fuzzy knowledge representation, International Journal of Innovative Computing, Information and Control 7 (7A) (2011) 3709–3724.
[29] Q. Wu, R. Law, Complex system fault diagnosis based on a fuzzy robust wavelet support vector classifier and an adaptive Gaussian particle swarm optimization, Information Sciences 180 (23) (2010) 4514–4528.
[30] L.A. Zadeh, Fuzzy Logic Technology and Their Application, IEEE Publications, 1994.