Fuzzy Membrane Computing: Theory and Applications

T. Wang, G. Zhang, M.J. Pérez-Jiménez

Tao Wang, Gexiang Zhang*

School of Electrical Engineering
Southwest Jiaotong University, Chengdu, 610031, China
wangatao2005@163.com, zhgxdylan@126.com
*Corresponding author: zhgxdylan@126.com

Mario J. Pérez-Jiménez

Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla, Sevilla, 41012, Spain
marper@us.es

Abstract: Fuzzy membrane computing is a newly developed and promising research direction in the area of membrane computing that aims at exploring the complex interaction between membrane computing and fuzzy theory. This paper provides a comprehensive survey of theoretical developments and various applications of fuzzy membrane computing, and sketches future research lines. The theoretical developments are reviewed from the aspects of uncertainty processing in P systems, fuzzification of P systems and fuzzy knowledge representation and reasoning. The applications of fuzzy membrane computing are mainly focused on fuzzy knowledge representation and fault diagnosis. An overview of different types of fuzzy P systems, differences between spiking neural P systems and fuzzy reasoning spiking neural P systems and newly obtained results on these P systems are presented.

Keywords: fuzzy membrane computing, fuzzy set, multi-fuzzy set, membrane computing, fuzzy reasoning spiking neural P systems, trapezoidal fuzzy number, linguistic term.

1 Introduction

Membrane computing (MC), introduced by Gh. Păun in [1], is an attractive research field of computer science aiming at abstracting computing models, called membrane systems or P systems, from the structures and functioning of living cells, as well as from the way the cells are organized in tissues or higher order structures. Since then, the MC research has been continuously and rapidly progressing [2], [3]. There are, basically, three main types of P systems: cell-like P systems, tissue-like P systems and neural-like P systems [3]. For all cases, a P system consists of three basic elements: membrane structure, multisets of objects and evolution rules. In recent years, the research on neural-like P systems mainly focused on spiking neural P systems (SN P systems), which were introduced in [4] being a type of P systems inspired by the neurophysiological behavior of neurons sending electrical impulses (spikes) along axons from presynaptic neurons to postsynaptic neurons in a distributed and parallel manner. Recently, SN P systems have become a hot topic in membrane computing [2, 5]- [25].

A fuzzy set is a class of objects with a continuum of grades of membership and is characterized by a membership (characteristic) function which assigns to each objects a grade of membership ranging in [0, 1]. The theory of fuzzy sets was proposed by Zadeh in [26] as an extension of the classical notion of set. Since then fuzzy set theory has been studied extensively over the past several decades years. Most of the early interests in fuzzy set theory pertained to representing uncertainty in human cognitive processes [27]. Fuzzy set theory is now combined with other

methods and applied to problems in engineering, business, medical and related health sciences, the natural sciences and so on [27]- [34].

P systems are models of computation inspired by the structure and functioning of cells in living organisms. Since uncertainty is an inherent property of all living systems, an increasing interest in the development of uncertain mathematical approaches to membrane computing is emerging due to three main reasons: keeping characteristics of P systems closer to the non-crisp behavior of real cells; the development of new formal computational models dealing with fuzzy information; the possibility of applying P systems to model real biological processes, where handling uncertainty, errors and approximations is necessary. Păun mentioned this in his first list of open problems in membrane computing for the development of approximate mathematical approaches in [35] and this topic continued to be discussed in [36]- [38].

A first contribution to this research line was given in [39], by extending the classical model to several probabilistic ones. Several possible rough set based mathematical models of uncertainty that could be used in membrane computing were further discussed in [40], [41]. Meanwhile, several fuzzy approaches based on fuzzy set theory have been introduced [12]- [25] [42]- [48]. The basic idea of fuzzifying P systems is the substitution of one or all ingredients in a P system with their fuzzy counterparts to obtain fuzzy P systems. Fuzzy P systems possess the same structure as crisp P systems but admit the association of numerical fuzzy values to elements in the membranes. Up to now, several aspects from fuzzy theory has been introduced into cell-like P systems [42]- [48] and SN P systems [12]- [25]. Fuzzy cell-like P systems mainly focused on theoretical research while fuzzy reasoning spiking neural P systems (FRSN P systems) concentrated on both theoretical and application researches. To date, five types of FRSN P systems were proposed for knowledge representation and reasoning as well as used for fault diagnosis. These FRSN P systems are fuzzy reasoning spiking neural P systems with real numbers (rFRSN P systems), fuzzy reasoning spiking neural P systems with linguistic terms (lFRSN P systems), adaptive fuzzy reasoning spiking neural P systems with real numbers (AFRSN P systems), weighted fuzzy reasoning spiking neural P systems (WFRSN P systems) and fuzzy reasoning spiking neural P systems with trapezoidal fuzzy numbers (tFRSN P systems).

Fuzzy membrane computing (FMC) has well established theoretical foundation and real world applications. To outline the work in the past twelve years on fuzzy membrane computing, this paper reviews and summarizes theoretical developments and various applications of fuzzy membrane computing. This comprehensive survey provides an overview of newly developed and promising research lines in the area of membrane computing as well as lists some promising research topics. As such, this work will be of interest to advance in the research line of fuzzy membrane computing, as well as to members of the membrane computing and fuzzy theory communities, specially newcomers.

The remainder of this paper is organized as follows. Section 2 provides an overview of theoretical achievements of FMC. The overview of the application development is outlined in Section 3. Finally, some conclusions and possible further developments are discussed.

2   Theoretical development

This section will present theoretical developments of FMC on the aspects of handling uncertainty in P systems, fuzzification of P systems and fuzzy knowledge representation and reasoning.

2.1   Uncertainty processing in P systems

MC is a formal computational model that rewrites multisets of objects within a spatial structure inspired by membrane structures and functioning of living cells, as well as from the way

the cells are organized in tissues or higher order structures and expresses biochemical processes taking place inside cells by evolution rules. Features of "exact" membrane computing developed so far are summarized as follows: objects used in computations are exact copies of reactants involved in biochemical reactions modeled by the rules; an application of a given rule always yields exact copies of objects that the rule is assumed to produce. However, in real life, cells do not behave only in this "exact" way. Biochemical reactions may deal with inexact, uncertain and mutated copies of reactants involved in the cells, and errors may happen accompanying with a biochemical reaction. This means that since inaccuracy and uncertainty exist in real cells i.e. real cells show a non-crisp behavior, inaccuracy and uncertainty should be considered when MC models are established. In this way, actual objects produced when applying rules during computations, and the actual objects conforming the result of computations themselves, would not need to be exact copies of the reactants that are assumed to be used when applying the rules along the computations, but only approximate copies of these reactants.

In [42]- [44], a fuzzy symport/antiport membrane system that uses inexact copies of reactants in the transitions was proposed to handle this kind of inaccuracy and uncertainty. Then, universality of this fuzzy model of computation was proved. The key ingredients in this model are the use of fuzzy multisets in configurations, the endowment of evolution rules with threshold mappings that determine the degree of accuracy of objects to reactants in order to be affected by the rules, and an appropriate way of evaluating the content of the output membrane at the end of a halting computation. This is a first step towards the use of fuzzy methods to answer a question proposed by Păun in the last problem of his first list of open problems in membrane computing [35]. In [44], a fuzzy model of P systems in which the objects involved in computations were colored by means of a finite family of fuzzy sets, and several applications of this model in computational biology were discussed. In [42]- [44], fuzzy methods were used to cope with the possibility that the objects in the membranes were imperfect, approximate copies of the reactants involved in the reactions.

Moreover, in [45], a t-norm based approach for handling imprecision was proposed in P systems. In this work, a P system with vague boundaries in the t-norm approach was proposed and only string-objects were considered inside the membrane structure.

2.2 Fuzzification of P systems

Uncertainty is an inherent property of all living systems. Curiously enough, computational models inspired by biological systems do not take, in general, under consideration this essential aspect of living systems. P systems are computing models inspired by the structure and functioning of cells in living organisms. Thus, it is more than necessary to introduce uncertainty into P system models to fuzzify them due to the development of new formal computational paradigms dealing with fuzzy information and the possibility of applying P systems to model real biological processes, where handling uncertainty, errors and approximations is necessary.

In [46], fuzzy set theory and fuzzy logic were considered in the framework of P systems in order to deal with imprecise biological information. The motivation of this research was to use P systems for modeling the functioning of specific cellular structures and phenomena and, as a final goal, making P systems useful and relevant tools for biologists, and hence motivating further cooperation among scientists working in the areas of P systems and microbiology.

In [47], two variants of P systems with fuzzy components, P systems with fuzzy data and P systems with fuzzy multiset rewriting rules, were introduced. The basic idea behind this attempt to fuzzify P systems was the substitution of one or all ingredients of a P system with their fuzzy counterparts. In this work, the theory of multi-fuzzy sets was developed and the notion of a fuzzy multiset rewriting rule was presented to define P systems with the aforementioned fuzzy

components, fuzzy data and fuzzy multiset rewriting rules. By silently assuming that fuzzy data were not the result of some fuzzification process, P systems with fuzzy data were shown to be a step towards real hypercomputation, while P systems with fuzzy multiset rewriting rules were shown to be equivalent to fuzzy Turing machines. Besides, the idea of P systems with both fuzzy data and fuzzy multiset rewriting rules was briefly discussed.

In [41], general fuzzy membrane systems with general multi-fuzzy sets and their evolution rules were introduced by using general multi-fuzzy sets to fuzzification of multisets and P systems. This kind of fuzzy membrane systems was motivated by some practical applications in biochemistry and medical science, where weakness causing uncertainty of an occurrence of an object in a system was determined not only by the number of occurrence copies of that object, but also by a quality of these copies.

In [48], an orthogonal approach for the fuzzification of both multisets and hybrid sets was presented. In this work, L-multi-fuzzy and L-fuzzy hybrid sets were introduced to P systems and a general fuzzy P system with L-multi-fuzzy sets was proposed. Although replacing the multi-fuzzy sets from [41] with L-multi-fuzzy sets did not improve the computational power of the resulted P systems, P systems with L-multi-fuzzy sets might be quite useful in modeling living organisms.

2.3   Fuzzy knowledge representation

Fuzzy reasoning spiking neural P systems (FRSN P systems) are proposed to handle fuzzy knowledge [12]- [25]. Until now, five classes of FRSN P systems, rFRSN P systems, lFRSN P systems, AFRSN P systems, WFRSN P systems and tFRSN P systems, have been proposed to represent fuzzy knowledge. In what follows, we first summarize fuzzy production rules used for fuzzy knowledge representation. Subsequently, we survey these FRSN P systems with respect to their development and definitions, providing for each of them the corresponding models for the considered fuzzy production rules.

Fuzzy production rules

Fuzzy production rules are widely used in fuzzy knowledge representation [29]- [30]. Fuzzy production rules consist of five types: simple fuzzy production rules, composite fuzzy conjunctive rules in the antecedent, composite fuzzy conjunctive rules in the consequent, composite fuzzy disjunctive rules in the antecedent and composite fuzzy disjunctive rules in the consequent. Rules are discussed below:

A *simple fuzzy production rule* is of the form

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

where R_i indicates the ith fuzzy production rule; $\tau_i \in [0,1]$ represents its certainty factor; p_j and p_k represents two propositions, each of which has a fuzzy truth value in $[0,1]$. If fuzzy truth values of propositions p_j and p_k are α_j and α_k, respectively, then $\alpha_k = \alpha_j * \tau_i$.

A *composite fuzzy conjunctive rule in the antecedent* is of the form

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

where $k \geq 3$, R_i indicates the ith fuzzy production rule; $\tau_i \in [0,1]$ represents its certainty factor; $p_1, \cdots, p_{k-1}$ are propositions in the antecedent part of the rule; p_k is a proposition in the consequent part of the rule. If fuzzy truth values of propositions $p_1, \cdots, p_k$ are $\alpha_1, \cdots, \alpha_k$, respectively, then $\alpha_k = \min(\alpha_1, \ldots, \alpha_{k-1}) * \tau_i$.

A *composite fuzzy conjunctive rule in the consequent* is of the form

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

where $k \geq 3$, R_i indicates the ith fuzzy production rule; $\tau_i \in [0,1]$ represents its certainty factor; p_1 is a proposition in the antecedent part of the rule; $p_2, \cdots, p_k$ are propositions in the consequent part of the rule. If fuzzy truth values of proposition $p_1, \cdots, p_k$ is $\alpha_1, \cdots, \alpha_k$, then $\alpha_2 = \alpha_1 * \tau_i, \ldots, \alpha_k = \alpha_1 * \tau_i$.

A *composite fuzzy disjunctive rules in the antecedent* is of the form

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

where $k \geq 3$, R_i indicates the ith fuzzy production rule; $\tau_i \in [0,1]$ represents its certainty factor; $p_1, \cdots, p_{k-1}$ are propositions in the antecedent part of the rule; p_k is a proposition in the consequent part of the rule. If fuzzy truth values of propositions $p_1, \cdots, p_k$ are $\alpha_1, \cdots, \alpha_k$, respectively, then $\alpha_k = \max(\alpha_1, \ldots, \alpha_{k-1}) * \tau_i$.

A *composite fuzzy disjunctive rule in the consequent* is of the form

$$\text{Type 5 } R_i\text{: IF } p_1 \text{ THEN } p_2 \text{ or } \cdots \text{ or } p_k \text{ (CF}=\tau_i) \tag{5}$$

where R_i indicates the ith fuzzy production rule; $\tau_i \in [0,1]$ represents its certainty factor; p_1 is a proposition in the antecedent part of the rule; $p_2, \cdots, p_k$ are propositions in the consequent part of the rule. This type of rules is unsuitable for knowledge representation due to the fact that it does not make any specific implication [29]. Thus, this type of rules is not discussed here and will not be considered in the following sections.

rFRSN P systems

Fuzzy reasoning spiking neural P systems with real numbers (rFRSN P systems) were first introduced in [12] and further investigated in [13]- [15]. The definition of an rFRSN P system is described in Definition 1.

Definition 1. An rFRSN P system of degree $m \geq 1$ is a tuple $\Pi = (A, \sigma_1, \ldots, \sigma_m, syn, I, O)$, 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

   (a) $\alpha_i \in [0,1]$ is a real number representing the (potential) value of spike contained in neuron σ_i (also called pulse value);

   (b) $\tau_i \in [0,1]$ is a real number representing the truth value associated with neuron σ_i;

   (c) r_i is a firing/spiking rule contained in neuron σ_i, of the form $E/a^\alpha \to a^\beta$, where $\alpha, \beta \in [0,1]$, $E = a^n$ is the firing condition and n represents the number of input synapses from other neurons to this 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. For neuron σ_i, if its firing rule $E/a^\alpha \to a^\beta$ can be applied, then the neuron fires. This means its pulse value $\alpha > 0$ is consumed (removed) and it produces a spike with value β;

3 $syn \subseteq \{1, 2, \ldots, m\} \times \{1, 2, \ldots, m\}$ with $i \neq j$ for all $(i, j) \in syn$, $1 \leq i, j \leq m$ (*synapses between neurons*);

4 I and O are nonempty sets that represent input neuron set and output neuron set, respectively.

In order to use fuzzy production rules for fuzzy knowledge representation, we need map them into rFRSN P systems. In the following, we summarize rFRSN P systems for fuzzy production rules, where value τ_i of each rule neuron is assigned to the certainty factor of the fuzzy production rule associated with it.

A *simple fuzzy production rule* can be modeled by an rFRSN P system Π_1, as shown in Figure 1(a), being $\Pi_1 = (A, \sigma_i, \sigma_j, \sigma_k, syn, I, O)$, 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 *composite fuzzy conjunctive rule in the antecedent* can be modeled by an rFRSN P system Π_2, as shown in Figure 1(b), being $\Pi_2 = (A, \sigma_1, \sigma_2, \ldots, \sigma_k, \sigma_{k+1}, syn, I, O)$, where

(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 $E/a^\alpha \to a^\alpha$.

(3) σ_{k+1} is an "AND"-type 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 $\beta = \alpha * \tau_i$.

(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 *composite fuzzy conjunctive rule in the consequent* can be modeled by an rFRSN P system Π_3, as shown in Figure 1(c), being $\Pi_3 = (A, \sigma_1, \sigma_2, \ldots, \sigma_k, \sigma_{k+1}, syn, I, O)$, where

(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 $E/a^\alpha \to a^\alpha$.

(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 $E/a^\alpha \to a^\beta$, 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 *composite fuzzy disjunctive rule in the antecedent* can be modeled by an rFRSN P system Π_4, as shown in Figure 1(d), being $\Pi_4 = (A, \sigma_1, \sigma_2, \ldots, \sigma_k, \sigma_{k+1}, syn, I, O)$, where

(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 $E/a^\alpha \to a^\alpha$.

(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 $E/a^\alpha \to a^\beta$, 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) Π_1 for simple fuzzy production rules

(b) Π_2 for composite fuzzy conjunctive rules in the antecedent

(c) Π_3 for composite fuzzy conjunctive rules in the consequent

(d) Π_4 for composite fuzzy disjunctive rules in the antecedent
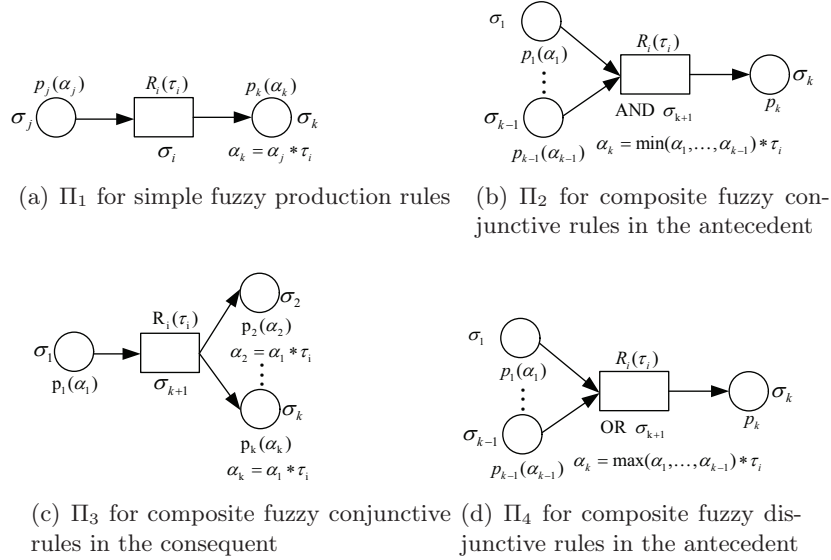
Figure 1: rFRSN P systems for fuzzy production rules

lFRSN P systems

The first version of fuzzy reasoning spiking neural P systems with linguistic terms (lFRSN P systems) was introduced in [16] and further expanded in [17]. The definition of a lFRSN P system is described in Definition 2.

Definition 2. An lFRSN P system of degree $m \geq 1$ is a tuple $\Pi = (O, P, R, Q, syn, in, out)$, where:

1. $O = \{a\}$ is the singleton alphabet (the object a is called spike);

2. $P = \{p_1, \ldots, p_k\}$ is a finite set of fuzzy propositions, where p_i represents the ith fuzzy proposition, $1 \leq i \leq k$;

3. $R = \{R_1, \ldots, R_n\}$ is a finite set of fuzzy productions rules, where R_i represents the ith fuzzy production rule, $1 \leq i \leq n$;

4. $Q = \{\sigma_1, \ldots, \sigma_m\}$ is a finite set of neurons, where $Q = Q_1 \cup Q_2$, $Q_1 \cap Q_2 = \emptyset$. $Q_1 = \{\sigma_1, \ldots, \sigma_k\}$ is a set of fuzzy proposition neurons and each proposition neuron i in Q_1 corresponds to a fuzzy proposition p_i in P, where $1 \leq i \leq k$. $Q_2 = \{\sigma_{k+1}, \ldots, \sigma_{k+n}\}$ is a set of rule neurons and each rule neuron $k + i$ in Q_2 corresponds to a fuzzy production rule R_i in R, where $1 \leq i \leq n$, $m = k + n$. Each neuron in Q has the form of $\sigma_i = (A_i, C_i, r_i)$, $1 \leq i \leq m$, where:

   (a) A_i is a linguistic term representing potential value contained in σ_i;

   (b) For proposition neurons in Q_1, C_i is ignored; for rule neurons in Q_2, C_i is a linguistic term representing the certainty factor of the corresponding fuzzy production rule of ith neuron;

   (c) r_i is a firing/spiking rule contained in neuron σ_i, of the form $a^\mu \to a^\nu$, where μ and ν are linguistic terms representing potential values of neurons. For rule neurons, μ is logical "and" (denoted by $\otimes$) or "or" (denoted by $\oplus$) of all inputs received by this neuron, and $\nu = \mu \otimes C_i$;

5. $syn \subseteq \{1, 2, \ldots, m\} \times \{1, 2, \ldots, m\}$ with $i \neq j$ for all $(i, j) \in syn$, $1 \leq i, j \leq m$ (synapses between neurons);

6. in and out are input neuron set and output neuron set, respectively.

In what follows, we summarize lFRSN P system models for rules *Type 1* to *Type 4*, as shown in Figure 2.
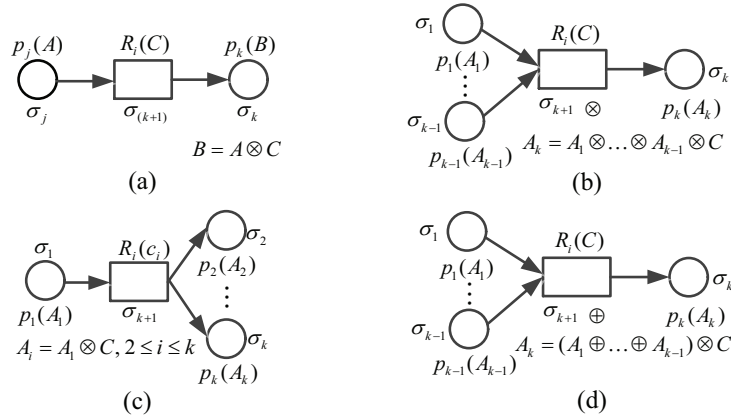

Figure 2: lFRSN P system models for fuzzy production rules. (a) *Type 1*; (b) *Type 2*; (c) *Type 3*; (d) *Type 4*.

Type 1 R_i: IF p_j THEN p_k (CF=C). The fuzzy truth value of proposition p_k is $B = A \otimes C$, where $A_\alpha = [a_1^\alpha, a_2^\alpha]$, $C_\alpha = [c_1^\alpha, c_2^\alpha]$, $\alpha \in [0, 1]$, i.e., $B = \int_0^1 \alpha[a_1^\alpha \wedge c_1^\alpha, a_2^\alpha \wedge c_2^\alpha]$.

Type 2 R_i: IF p_1 and $\cdots$ and p_{k-1} THEN p_k (CF=C). The fuzzy truth value of proposition p_k is $A_k = A_1 \otimes \ldots \otimes A_{k-1} \otimes C$, where $[a_{i1}^\alpha, a_{i2}^\alpha]$ is α-cut of A_i, $1 \leq i \leq k - 1$, $\alpha \in [0, 1]$, i.e., $A_k = \int_0^1 \alpha[a_{11}^\alpha \wedge a_{(k-1)1}^\alpha \wedge \ldots \wedge c_1^\alpha, a_{12}^\alpha \wedge \ldots \wedge a_{(k-1)2}^\alpha \wedge c_2^\alpha]$.

Type 3 R_i: IF p_1 THEN p_3 and $\cdots$ and p_k (CF=C). The fuzzy truth value of propositions $p_2, \ldots, p_k$ are identical, i.e., $A_i = A_1 \otimes C$, $2 \leq i \leq k$, where $[a_1^\alpha, a_2^\alpha]$ is α-cut of A_1, $\alpha \in [0, 1]$, i.e., $A_i = \int_0^1 \alpha[a_1^\alpha \wedge c_1^\alpha, a_2^\alpha \wedge c_2^\alpha]$, $2 \leq i \leq k$.

Type 4 R_i: IF p_1 or p_2 or $\cdots$ or p_{k-1} THEN p_k (CF=C). The fuzzy truth value of proposition p_k is $A_k = (A_1 \otimes \ldots \oplus A_{k-1}) \otimes C$, where $[a_{i1}^\alpha, a_{i2}^\alpha]$ is α-cut of A_i, $1 \le i \le k-1$, $\alpha \in [0,1]$, i.e., $A_k = \int_0^1 \alpha[(a_{11}^\alpha \vee \ldots \vee a_{(k-1)1}^\alpha) \wedge c_1^\alpha, (a_{12}^\alpha \vee \ldots \vee a_{(k-1)2}^\alpha) \wedge c_2^\alpha]$.

AFRSN P systems

Adaptive fuzzy reasoning spiking neural P systems (AFRSN P systems) were first introduced in [20] and further investigated in [21]. The definition of an AFRSN P system is described in Definition 3 [21].

Definition 3. An AFRSN P system (*of degree $m \ge 1$*) is a tuple $\Pi = (A, N_p, N_r, syn, I, O)$, where

- $A=\{a\}$ is the singleton alphabet (the object a is called spike);

- $N_p = \{\sigma_{p1}, \ldots, \sigma_{pm}\}$ is proposition neuron set, where proposition neuron σ_{pi} expresses the ith proposition of weighted fuzzy production rules, $1 \le i \le m$. $\sigma_{pi} = (\alpha_i, \omega_i, \lambda_i, r_i)$, where

  - $\alpha_i \in [0,1]$ is the pulse value contained in proposition neuron σ_{pi}. α_i is used to express fuzzy truth value of the proposition associated with proposition neuron σ_{pi};

  - $\omega_i = (\omega_{i1}, \ldots, \omega_{is_i})$ expresses the output weight vector of neuron σ_{pi}, where $\omega_{ij} \in [0,1]$ is the weight on jth output synapse of the neuron, $1 \le j \le s_i$, and s_i is the number of all output synapses of the neuron;

  - r_i is a firing/spiking rule, of the form $E/a^\alpha \to a^\alpha$, $\alpha \in [0,1]$. $E = \{\alpha \ge \lambda_i\}$ is the firing condition, i.e., if $\alpha \ge \lambda_i$, then the firing rule will be enabled, where $\lambda_i \in [0,1)$ is called the firing threshold;

- $N_r = \{\sigma_{r1}, \ldots, \sigma_{rn}\}$ is rule neuron set, where rule neuron σ_{ri} expresses the ith weighted fuzzy production rule, $1 \le i \le n$. $\sigma_{ri} = (\alpha_i, \gamma_i, \tau_i, r_i)$, where

  - $\alpha_i \in [0,1]$ is the potential value (spike) contained in rule neuron σ_{ri};

  - $\gamma_i \in [0,1]$ is the certain factor, which represents the strength of belief of the weighted fuzzy production rule associated with rule neuron σ_{ri}. α_i is also the weight on output synapse (arc) of the neuron;

  - r_i is a firing/spiking rule, of the form $E/a^\alpha \to a^\beta$, where $\alpha, \beta \in [0,1]$. $E = \{\alpha \ge \tau_i\}$ is the firing condition, i.e., if $\alpha \ge \tau_i$, then the firing rule will be enabled, where $\tau_i \in [0,1)$ is called the firing threshold;

- $syn \subseteq (N_p \times N_r) \bigcup (N_r \times N_p)$ indicates synapses between proposition neurons and rule neurons. Note that there are no synapse connections between any two proposition neurons or between any two rule neurons;

- $I, O \subseteq N_p$ are input neuron set and output neuron set, respectively.

The motivation of proposing AFRSN P systems is to model weighted fuzzy production rules and the following three types are concerned. AFRSN P systems models for weighted fuzzy production rules are summarized as follows, as shown in Figure 3.

Type 1 R_i: IF p_1 THEN p_2 ($CF = \gamma$), τ, ω;

Type 2 R_i: IF p_1 AND p_2 AND $\cdots$ AND p_n THEN p_{n+1} ($CF = \gamma$), τ, $\omega_1, \ldots, \omega_n$;

Type 3 R_i: IF p_1 OR p_2 OR $\cdots$ OR p_n THEN p_{n+1} ($CF = \gamma$), τ, $\omega_1, \ldots, \omega_n$.

According to dynamic firing mechanism of AFRSN P systems, the computing ways of spikes for AFRSN P systems of these fuzzy production rules are described as follows [20].
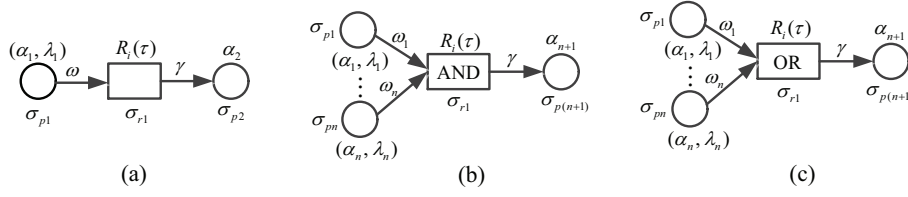
Figure 3: AFRSN P systems of weighted fuzzy production rules of three types (a) type 1. (b) type 2. (c) type 3.

- Type 1: $\alpha_2 = \begin{cases} \alpha_1 \cdot \gamma, & \text{if } \alpha_1 \geq \tau \\ 0, & \text{if } \alpha_1 < \tau \end{cases}$

- Type 2: $\alpha_{n+1} = \begin{cases} \left(\sum\limits_{i=1}^{n} \alpha_i \cdot \omega_i \right) \cdot \gamma, & \text{if } \left(\sum\limits_{i=1}^{n} \alpha_i \cdot \omega_i \right) \geq \tau \\ 0, & \text{if } \left(\sum\limits_{i=1}^{n} \alpha_i \cdot \omega_i \right) < \tau \end{cases}$

- Type 3: $\alpha_{n+1} = \begin{cases} \max(\alpha_j \cdot \gamma_j), & \text{if } \alpha_j \geq \tau_j, j \in J \\ 0, & \text{if } \alpha_j < \tau_j, j = 1, 2 \ldots, n \end{cases}$

WFRSN P systems

Weighted fuzzy reasoning spiking neural P systems (WFRSN P systems) were first introduced in [18] and further investigated in [19]. The definition of a WFRSN P system is described in Definition 4.

Definition 4. A WFRSN P system of degree $m \geq 1$ is a tuple $\Pi = (O, \sigma_1, \ldots, \sigma_m, syn, in, out)$, where:

1. $O = \{a\}$ is a singleton alphabet (a is called spike);

2. $\sigma_1, \ldots, \sigma_m$ are neurons, of the form $\sigma_i = (\theta_i, c_i, \overrightarrow{\omega_i}, \lambda_i, r_i), 1 \leq i \leq m$, where:

   (a) θ_i is a real number in $[0, 1]$ representing the potential value of spikes (i.e. value of electrical impulses) contained in neuron σ_i;

   (b) c_i is a real number in $[0, 1]$ representing the truth value associated with neuron σ_i;

   (c) $\overrightarrow{\omega_i} = (\omega_{i1}, \ldots, \omega_{iN_i})$ is a real number vector in $(0, 1]$ representing the output weight vector of neuron σ_i, where ω_{ij} $(1 \leq j \leq N_i)$ represents the weight on jth output arc (synapse) of neuron σ_i and N_i is a natural number representing the number of synapses starting from neuron σ_i.

   (d) λ_i is a real number in $[0, 1)$ representing the firing threshold of neuron σ_i;

   (e) r_i represents a firing (spiking) rule contained in neuron σ_i with the form $E/a^\theta \rightarrow a^\beta$, where θ and β are real numbers in $[0, 1]$, $E = \{a^n, \theta \geq \lambda_i\}$ is the firing condition. The firing condition means that if and only if neuron σ_i receives at least n spikes and the potential value of spikes is with $\theta \geq \lambda_i$, then the firing rule contained in the neuron can be applied, otherwise, the firing rule cannot be applied;

3. $syn \subseteq \{1, \ldots, m\} \times \{1, \ldots, m\}$ with $i \neq j$ for all $(i, j) \in syn, 1 \leq i, j \leq m$; that is, syn provides a (weighted) directed graph whose set of nodes is $\{1, \ldots, m\}$;

4. $in, out \subseteq \{1, \ldots, m\}$ indicate the input neuron set and the output neuron set of Π, respectively.

In what follows, we describe fault diagnosis production rules and their WFRSN P system models, as shown in Figure 4 [19], [49].
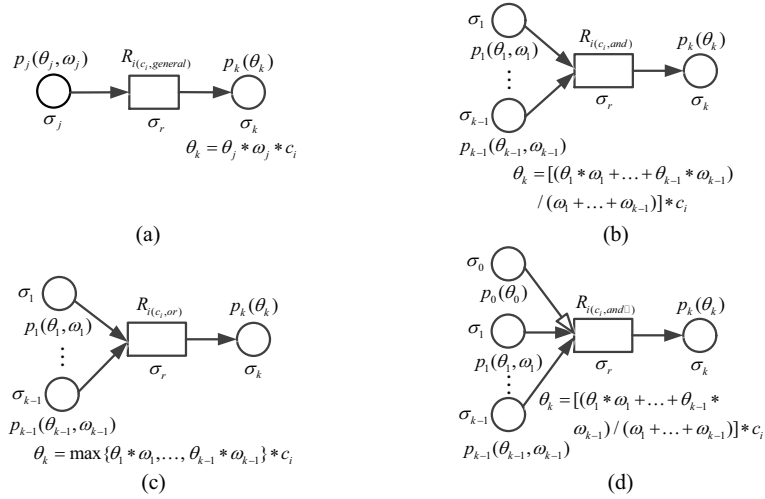

Figure 4: WFRSN P system models for fault diagnosis production rules in TPSSs. (a) *Type 1*; (b) *Type 2*; (c) *Type 3*; (d) *Type4*.

Type 1 (Simple Rules) R_i: IF $p_j(\theta_j)$ THEN $p_k(\theta_k)$ (CF $= c_i$), where p_j and p_k are propositions, c_i is a real number in [0, 1] representing the certainty factor of rule R_i, θ_j and θ_k are real numbers in [0, 1] representing the truth values of p_j and p_k, respectively. The weight of proposition p_j is ω_j, where $\omega_j = 1$ because there is only one proposition in the antecedent of this kind of rules. The truth values of p_k is $\theta_k = \theta_j * \omega_j * c_i = \theta_j * c_i$.

Type 2 (Compound And Rules) R_i: IF $p_1(\theta_1)$ and $\ldots$ and $p_{k-1}(\theta_{k-1})$ THEN $p_k (\theta_k)$ (CF $= c_i$), where $p_1, \ldots, p_k$ are propositions, c_i is a real number in [0, 1] representing the certainty factor of rule R_i, $\theta_1, \ldots, \theta_k$ are real numbers in [0, 1] representing the truth values of $p_1, \ldots, p_k$, respectively. The weights of propositions $p_1, \ldots, p_{k-1}$ are $\omega_1, \ldots, \omega_{k-1}$, respectively. The truth values of p_k is $\theta_k = [(\theta_1 * \omega_1 + \ldots + \theta_{k-1} * \omega_{k-1})/(\omega_1 + \ldots + \omega_{k-1})] * c_i$.

Type 3 (Compound Or Rules) R_i: IF $p_1(\theta_1)$ or $\ldots$ or $p_{k-1}(\theta_{k-1})$ THEN $p_k (\theta_k)$ (CF $= c_i$), where $p_1, \ldots, p_k$ are propositions, c_i is a real number in [0, 1] representing the certainty factor of rule R_i, $\theta_1, \ldots, \theta_k$ are real numbers in [0, 1] representing the truth values of $p_1, \ldots, p_k$, respectively. The weights of propositions $p_1, \ldots, p_{k-1}$ are $\omega_1, \ldots, \omega_{k-1}$, respectively. The truth values of p_k is $\theta_k = max\{\theta_1 * \omega_1, \ldots, \theta_{k-1} * \omega_{k-1}\} * c_i$.

Type 4 (Conditional And Rules) R_i: WHEN $p_0(\theta_0)$ is true, IF $p_1(\theta_1)$ and $\ldots$ and $p_{k-1}(\theta_{k-1})$ THEN $p_k (\theta_k)$ (CF $= c_i$), where $p_0, \ldots, p_k$ are propositions, c_i is a real number in [0, 1] representing the certainty factor of rule R_i, $\theta_0, \ldots, \theta_k$ are real numbers in [0, 1] representing the truth values of $p_0, \ldots, p_k$, respectively. The proposition p_0 is used to judge whether the reasoning condition of rule R_i is satisfied and its truth value θ_0 is not used in reasoning process. Thus, the weight of θ_0 is not considered in the model. The weights of propositions $p_1, \ldots, p_{k-1}$ are $\omega_1, \ldots, \omega_{k-1}$, respectively. The truth values of p_k is $\theta_k = [(\theta_1 * \omega_1 + \ldots + \theta_{k-1} * \omega_{k-1})/(\omega_1 + \ldots + \omega_{k-1})] * c_i$.

tFRSN P systems

The first version of fuzzy reasoning spiking neural P systems with trapezoidal fuzzy numbers (tFRSN P systems) was introduced in [22] and further expanded in [23]- [25]. The definition of a tFRSN P system is described in Definition 5.

Definition 5. A tFRSN P system with trapezoidal fuzzy numbers of degree $m \geq 1$ is a tuple $\Pi = (O, \sigma_1, \ldots, \sigma_m, syn, in, out)$, where:

1. $O = \{a\}$ is a singleton alphabet (a is called spike);

2. $\sigma_1, \ldots, \sigma_m$ are neurons of the form $\sigma_i = (\theta_i, c_i, r_i), 1 \leq i \leq m$, where

    (a) θ_i is a trapezoidal fuzzy number in [0, 1] representing the potential value of spikes (i.e., the value of electrical impulses) contained in neuron σ_i;

    (b) c_i is a trapezoidal fuzzy number in [0, 1] representing the fuzzy truth value corresponding to neuron σ_i;

    (c) r_i represents a firing (spiking) rule associated with neuron σ_i of the form $E/a^\theta \rightarrow a^\beta$, where θ and β are trapezoidal fuzzy numbers in [0, 1], $E = a^s$ is the firing condition, it means that the spiking rule contained in neuron σ_i, can be applied if and only if neuron σ_i contains at least s spikes, otherwise, the firing rule cannot be applied;

3. $syn \subseteq \{1, 2, \ldots, m\} \times \{1, 2, \ldots, m\}$ with $i \neq j$ for all $(i, j) \in syn, 1 \leq i, j \leq m$, is a directed graph of synapses between the linked neurons;

4. $in, out \in \{1, 2, \ldots, m\}$ indicate the input neuron set and the output neuron set of Π, respectively.

A *trapezoidal fuzzy number* can be parameterized by a 4-tuple $\tilde{A}=(a_1, a_2, a_3, a_4)$, as shown in Fig. 5, where a_1, a_2, a_3 and a_4 are real numbers such that $a_1 < a_2 < a_3 < a_4$, which are the four horizontal axis values of the trapezoid. The membership function $\mu_{\tilde{A}}(x)$ of the trapezoidal fuzzy number $\tilde{A}$ is defined as follows.

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, & x \leq a_1 \\ \frac{x-a_1}{a_2-a_1}, & a_1 < x \leq a_2 \\ 1, & a_2 < x \leq a_3 \\ \frac{a_4-x}{a_4-a_3}, & a_3 < x \leq a_4 \\ 0, & x > a_4 \end{cases} \tag{6}$$
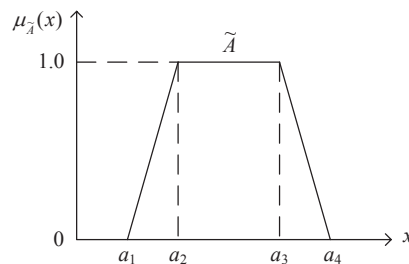

Figure 5: A trapezoidal fuzzy number.

Let $\tilde{A}$ and $\tilde{B}$ be two trapezoidal fuzzy numbers, $\tilde{A} = (a_1, a_2, a_3, a_4)$ and $\tilde{B} = (b_1, b_2, b_3, b_4)$. The arithmetic operations of the trapezoidal fuzzy numbers $\tilde{A}$ and $\tilde{B}$ are listed as follows. More operations can be seen in [25], [29].

1. Addition $\oplus$: $\tilde{A} \oplus \tilde{B} = (a_1, a_2, a_3, a_4) \oplus (b_1, b_2, b_3, b_4) = (a_1 + b_1, a_2 + b_2, a_3 + b_3, a_4 + b_4)$;

2. Subtraction $\ominus$: $\tilde{A} \ominus \tilde{B} = (a_1, a_2, a_3, a_4) \ominus (b_1, b_2, b_3, b_4) = (a_1 - b_1, a_2 - b_2, a_3 - b_3, a_4 - b_4)$;

3. Multiplication $\otimes$: $\tilde{A} \otimes \tilde{B} = (a_1, a_2, a_3, a_4) \otimes (b_1, b_2, b_3, b_4) = (a_1 \times b_1, a_2 \times b_2, a_3 \times b_3, a_4 \times b_4)$;

4. Division $\oslash$: $\tilde{A} \oslash \tilde{B} = (a_1, a_2, a_3, a_4) \oslash (b_1, b_2, b_3, b_4) = (a_1/b_1, a_2/b_2, a_3/b_3, a_4/b_4)$.

Four logic operations are list as follows, where A and B are trapezoidal fuzzy numbers, and a, b are real numbers [25].

1. *Minimum operator* $\wedge$: $a \wedge b = min(a, b)$;

2. *Maximum operator* $\vee$: $a \vee b = max(a, b)$;

3. *and* $\wedge$: $A \wedge B = (a_1, a_2, a_3, a_4) \wedge (b_1, b_2, b_3, b_4) = ((a_1 \wedge b_1), (a_2 \wedge b_2), (a_3 \wedge b_3), (a_4 \wedge b_4))$;

4. *or* $\vee$: $A \vee B = (a_1, a_2, a_3, a_4) \vee (b_1, b_2, b_3, b_4) = ((a_1 \vee b_1), (a_2 \vee b_2), (a_3 \vee b_3), (a_4 \vee b_4))$.

In addition, a scalar multiplication operation is list as follows, where A is a trapezoidal fuzzy number and b is a real number [25].

Scalar Multiplication: $bA = b(a_1, a_2, a_3, a_4) = (ba_1, ba_2, ba_3, ba_4)$.

In what follows, we summarize tFRSN P system models for rules *Type 1* to *Type 4*, as shown in Figure 6. In the following description, R_i $(i = 1, \ldots, N_r)$ is the ith fuzzy production rule, N_r represents the number of fuzzy production rules, c_i is a trapezoidal fuzzy number in $[0, 1]$ representing the certainty factor of R_i, p_j $(1 \le j \le N_p)$ is the jth proposition appearing in the antecedent or consequent part of R_i, N_p represents the number of proportions, and θ_j is a trapezoidal fuzzy number in $[0, 1]$ representing the fuzzy truth value of proposition p_j.
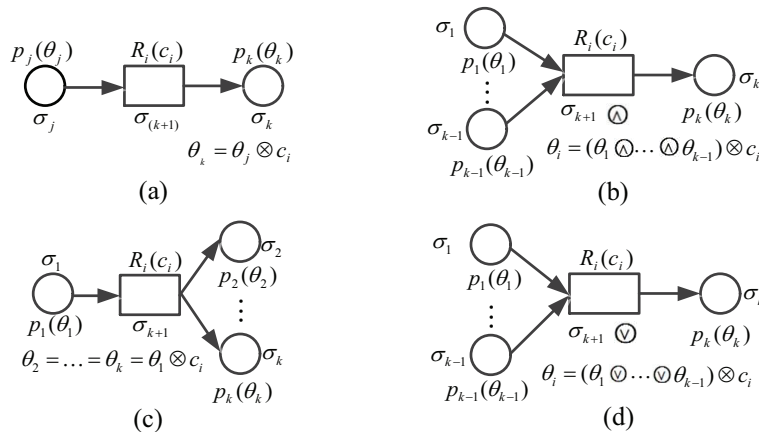

Figure 6: tFRSN P system models for fuzzy production rules. (a) *Type 1*; (b) *Type 2*; (c) *Type 3*; (d) *Type 4*.

Type 1: $R_i(c_i) : p_j(\theta_j) \rightarrow p_k(\theta_k)$ $(1 \le j, k \le N_p)$. The fuzzy truth value of the proposition p_k is $\theta_k = \theta_j \otimes c_i$.

Type 2: $R_i(c_i) : p_1(\theta_1) \bigwedge \ldots \bigwedge p_{k-1} (\theta_{k-1}) \rightarrow p_k (\theta_k)$. The fuzzy truth value of the proposition p_k is $\theta_k = (\theta_1 \bigwedge \ldots \bigwedge \theta_{k-1}) \otimes c_i$.

Type 3: $R_i(c_i) : p_1(\theta_1) \rightarrow p_2(\theta_2) \bigwedge \ldots \bigwedge p_k (\theta_k)$. The fuzzy truth values of the propositions $p_2, p_3, \ldots, p_k$ are identical, i.e., $\theta_2 = \theta_3 = \ldots = \theta_k = \theta_1 \otimes c_i$.

Type 4: $R_i(c_i) : p_1(\theta_1) \bigvee \ldots \bigvee p_{k-1} (\theta_{k-1}) \rightarrow p_k(\theta_k)$. The fuzzy truth value of the proposition p_k is $\theta_k = (\theta_1 \bigvee \ldots \bigvee \theta_{k-1}) \otimes c_i$.

The fuzzy truth values of these propositions appearing in the fuzzy production rules and the certainty factor of each fuzzy production rule can also be described by using linguistic terms, which are represented by the trapezoidal fuzzy numbers shown in Table 1.

Linguistic Terms	Trapezoidal Fuzzy Numbers
absolutely-false (AF)	(0, 0, 0, 0)
very-low (VL)	(0, 0, 0.02, 0.07)
low (L)	(0.04, 0.1, 0.18, 0.23)
medium-low (ML)	(0.17, 0.22, 0.36, 0.42)
medium (M)	(0.32, 0.41, 0.58, 0.65)
medium-high (MH)	(0.58, 0.63, 0.80, 0.86)
high (H)	(0.72, 0.78, 0.92, 0.97)
very-high (VH)	(0.975, 0.98, 1, 1)
absolutely-high (AH)	(1, 1, 1, 1)

Table 1: Linguistic terms and their corresponding trapezoidal fuzzy numbers

2.4 Fuzzy knowledge reasoning

Fuzzy reasoning spiking neural P systems (FRSN P systems) are proposed to handle fuzzy knowledge. In what follows, we will summarize fuzzy reasoning algorithms for the following FRSN P systems classes: rFRSN P systems, AFRSN P systems, WFRSN P systems and tFRSN P systems.

Fuzzy reasoning based on rFRSN P systems

A fuzzy reasoning algorithm (FRA) based on rFRSN P systems was proposed in [12]. The goal of FRA is to reason out the fuzzy truth values of unknown fuzzy propositions (proposition neurons) from known fuzzy propositions (input neurons). These unknown fuzzy propositions are associated with output neurons. Suppose all fuzzy production rules in a fuzzy diagnosis knowledge base have been modeled by an rFRSN P system model Π. The model Π consists of m neurons consisting of n proposition neurons and k rule neurons (AND type neurons and OR type neurons), where $m = n + k$.

A description of an FRA for an rFRSN P system is shown below. For details about the involved parameter vectors, matrices and multiplication operations, please see [12].

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

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

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

(3) Process the firing of rule neurons.

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

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

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

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).

Fuzzy reasoning based on AFRSN P systems

The fuzzy reasoning algorithm for AFRSN P systems is about the fuzzy reasoning process of above weighted fuzzy production rules. Let $Pcurrent = \{\sigma_{pi}|\sigma_{pi} \in N_p, \alpha_i > 0\}$ be a set of current enabled proposition neurons. If a neuron $\sigma_{pi} \in Pcurrent$, then it fires. Let $Rcurrent = \{\sigma_{rj}|\sigma_{rj} \in N_r, \alpha_i > \tau_j\}$ be a set of current enabled rule neurons. Likewise, if a neuron $\sigma_{rj} \in Rcurrent$, then it fires. A fuzzy reasoning algorithm for AFRSN P systems can be summarized as follows.

INPUT: Certainty factors of a set of antecedent propositions, which correspond to I of an AFSN P system.
OUTPUT: The fuzzy truth values of propositions associated with the neurons in O.
Step 1) $Pcurrent := I$, $Rcurrent := \{\}$, $P := Np$, $R := Nr$;
Step 2) Let $t = 0$, where t represents the reasoning step;
Step 3) Compute the outputs of current enabled proposition neurons in $Pcurrent$;
Step 4) Find current enabled rule neurons $Rcurrent$ form R;
Step 5) Compute the outputs of current enabled proposition neurons in $Rcurrent$;
Step 6) $P := P - Pcurrent$, $R := R - Rcurrent$;
Step 7) Find current enabled proposition neurons $Pcurrent$ form P;
Step 8) If $P = \{\}$ or $R = \{\}$ (computation halts), the reasoning
results are obtained; otherwise, $t = t + 1$, go to Step 3).

A weight learning algorithm for AFRSN P systems is summarized as follows.

INPUT: Training data set D, $m = |D|$, learning rate delta.
OUTPUT: Weights $(w_1, w_2, ..., w_n)$, where n is the number of weights.
Step 1) Select a set of initial weights;
Step 2) Let $t = 1$, where t represents the reasoning step;
Step 3) Update the weights $(w_1, w_2, ..., w_n)$, using Widrow-Hoff learning law with learning rate delta;
Step 4) If $t > m$ (computation halts) and m represents the number of proposition neurons, the reasoning results are obtained; otherwise, $t = t + 1$, go to Step 3).

Fuzzy reasoning based on WFRSN P systems

A weighted matrix fuzzy reasoning algorithm (WMFRA) for WFRSN P systems can be summarized as follows [19]. For details about the involved parameter vectors, matrices and mul-

tiplication operations, please see [19].

INPUT: The fuzzy truth values of the propositions corresponding to the input proposition neurons.

OUTPUT: The fuzzy truth values of the propositions corresponding to the output proposition neurons.

Step 1) Let $g = 0$ be the reasoning step;

Step 2) Set initial values of $\boldsymbol{W}_{r1}$, $\boldsymbol{W}_{r2}$, $\boldsymbol{W}_{r3}$, $\boldsymbol{W}_p$, $\boldsymbol{\lambda}_p$, $\boldsymbol{\lambda}_r$, $\mathbf{C}$, and the termination condition $\mathbf{0}_1 = (0, \overset{(t)}{...}, 0)^T$. The initial values of $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ are set to $\boldsymbol{\theta}_g = (\theta_{1g}, \theta_{2g}, \ldots, \theta_{sg})$ and $\boldsymbol{\delta}_g = (\delta_{1g}, \delta_{2g}, \ldots, \delta_{tg})$, respectively;

Step 3) g is increased by one;

Step 4) The firing condition of each input neuron ($g = 1$) or each proposition neuron ($g > 1$) is evaluated. If the condition $E = \{a^n, \theta_i \geq \lambda_{pi}, 1 \leq i \leq s\}$ is satisfied and there is a postsynaptic rule neuron, the neuron fires and transmits a spike to the next rule neuron;

Step 5) Compute the fuzzy truth value vector $\boldsymbol{\delta}_g$ according to (7);

$$\boldsymbol{\delta}_{g+1} = (\mathbf{W}_{r1}^T \otimes \boldsymbol{\theta}_g) + (\mathbf{W}_{r2}^T \oplus \boldsymbol{\theta}_g) + (\mathbf{W}_{r3}^T \odot \boldsymbol{\theta}_g) \tag{7}$$

Step 6) If $\boldsymbol{\delta}_g = \mathbf{0}_1$, the algorithm stops and outputs the reasoning results ;

Step 7) Evaluate the firing condition of each rule neuron. If the condition $E = \{a^n, \delta_j \geq \lambda_{rj}, 1 \leq j \leq t \}$ holds, the rule neuron fires and transmits a spike to the next proposition neuron;

Step 8) Compute the fuzzy truth value vector $\boldsymbol{\theta}_g$ according to (8). Go to Step 3).

$$\boldsymbol{\theta}_{g+1} = \mathbf{W}_p^T \odot (\mathbf{C} \otimes \boldsymbol{\delta}_{g+1}) \tag{8}$$

Fuzzy reasoning based on tFRSN P systems

A matrix-based fuzzy reasoning algorithm (MBFRA) for tFRSN P systems can be summarized as follows [25]. For details about the involved parameter vectors, matrices and multiplication operations, please see [25].

INPUT: The fuzzy truth values of the propositions corresponding to the input proposition neurons.

OUTPUT: The fuzzy truth values of the propositions corresponding to the output proposition neurons.

Step 1) Let $g = 0$ be the reasoning step;

Step 2) Set initial values of $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, $\boldsymbol{D}_3$, $\boldsymbol{E}$, $\boldsymbol{C}$ and the termination condition $\mathbf{0}_1 = (unknown, \overset{(t)}{...}, unknown)^T$. The initial values of $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ are set to $\boldsymbol{\theta}_g = (\theta_{1g}, \theta_{2g}, \ldots, \theta_{sg})$ and $\boldsymbol{\delta}_g = (\delta_{1g}, \delta_{2g}, \ldots, \delta_{tg})$, respectively;

Step 3) g is increased by one;

Step 4) The firing condition of each input neuron ($g = 1$) or each proposition neuron ($g > 1$) is evaluated. If the condition $E = a^s$ is satisfied and there is a postsynaptic rule neuron, the neuron fires and transmits a spike to the next rule neuron;

Step 5) Compute the fuzzy truth value vector $\boldsymbol{\delta}_g$ according to (9);

$$\boldsymbol{\delta}_g = (\boldsymbol{D}_1^T \odot \boldsymbol{\theta}_{g-1}) \oplus (\boldsymbol{D}_2^T \odot \boldsymbol{\theta}_{g-1}) \oplus (\boldsymbol{D}_3^T \circledast \boldsymbol{\theta}_{g-1}) \tag{9}$$

Step 6) If $\boldsymbol{\delta}_g = \mathbf{0}_1$, the algorithm halts and outputs the reasoning results;

Step 7) Evaluate the firing condition of each rule neuron. If the condition $E = a^s$ is satisfied,

the rule neuron fires and transmits a spike to the next proposition neuron;

Step 8) Compute the fuzzy truth value vector $\boldsymbol{\theta}_g$ according to (10). Go to Step 3).

$$\theta_g = \boldsymbol{E}^T \circledast (\boldsymbol{C} \odot \delta_g) \tag{10}$$

2.5   Comparisons between SN P systems and FRSN P systems

rFRSN P systems, lFRSN P systems, AFRSN P systems, WFRSN P systems and tFRSN P systems are collectively called FRSN P systems in this paper. FRSN P systems are novel graphical models for representing and reasoning fuzzy knowledge and information. How FRSN P systems are extended from SN P systems is described as follows and comparisons about neurons and spiking rules between SN P systems and FRSN P systems are shown in Table 2. It is worth pointing out that the spiking rule is collectively described with the form $E/a^\alpha \to a^\beta$ in FRSN P systems for convenient representation.

1. The content of a neuron is the potential value of spikes contained in this neuron instead of the number of spikes in SN P systems;

2. Each neuron in an FRSN P system associates with either a fuzzy proposition or a fuzzy production rule;

3. Each neuron contains only one spiking (firing) rule, with the form $E/a^\alpha \to a^\beta$, where $E = a^n$ is the firing condition and n represents the number of input synapses from other neurons to this 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;

4. The firing mechanism of neurons in FRSN P systems is described as follows. For the neuron σ_i, if its firing rule $E/a^\alpha \to a^\beta$ can be applied, then the neuron fires. This means its pulse value $\alpha > 0$ is consumed (removed) and it produces a spike with value β. Once a spike with value β is excited from neuron σ_i, all neurons σ_j with $(i, j) \in syn$ immediately receive the spike;

5. Different types of neurons are defined with different ways to handle spikes;

6. Time delay is ignored in an FRSN P system, thus all neurons are always open;

7. For AFRSN P systems and WFRSN P systems, their synapses have output weights.

3   Applications

This section will outline the application developments of fuzzy membrane computing including fuzzy knowledge representation and fault diagnosis of transformers, traction power supply systems and power transmission networks based on FRSN P systems. The description of the essentials of electrical power system fault diagnosis, please see [25] and principles of model-based fault diagnosis methods, please see [19].

P systems	Neurons			Spiking rules
	Type	Content	Expression	
SN P systems	one	number of spikes	computing space	$E/a^c \to a^p; d$
rFRSN P systems	proposition neuron rule neuron: and, or	pulse value (real numbers)	propositions, fuzzy production rules	$E/a^\alpha \to a^\beta$
lFRSN P systems	proposition neuron rule neuron	pulse value (linguistic terms)	propositions, fuzzy production rules	$E/a^\mu \to a^\nu$
AFRSN P systems	proposition neuron rule neuron	pulse value (real numbers)	propositions, fuzzy production rules	$E/a^\alpha \to a^\beta$
WFRSN P systems	proposition neuron rule neuron: general, and, or	pulse value (real numbers)	propositions, fuzzy production rules	$E/a^\alpha \to a^\beta$
tFRSN P systems	proposition neuron rule neuron: general, and, or	pulse value (trapezoidal fuzzy numbers)	propositions, fuzzy production rules	$E/a^\theta \to a^\beta$

Table 2: Comparisons about neurons and spiking rules between SN P systems and FRSN P systems

3.1 Fuzzy knowledge representation

Several kinds of FRSN P systems were proposed for fuzzy knowledge representation and reasoning. In this subsection, an example of fuzzy knowledge representation based on lFRSN P systems is summarized to show the effectiveness of FRSN P systems in representing fuzzy knowledge. Assume that there are seven fuzzy production rules in a rule set seven propositions represented by $p_1, p_2, p_3, p_4, p_5, p_6, p_7$ [16].

$R_1 : IF \ p_1 \ THEN \ p_2 \ (CF = almost \ certain)$
$R_2 : IF \ p_2 \ THEN \ p_3 \ (CF = pretty \ true)$
$R_3 : IF \ p_2 \ THEN \ p_4 \ (CF = rather \ true)$
$R_4 : IF \ p_1 \ THEN \ p_6 \ (CF = pretty \ true)$
$R_5 : IF \ p_6 \ THEN \ p_4 \ and \ p_5 \ (CF = very \ true)$
$R_6 : IF \ p_1 \ THEN \ p_7 \ (CF = almost \ certain)$
$R_7 : IF \ p_7 \ THEN \ p_4 \ (CF = sort \ of \ true)$

The fuzzy truth value *very true* of proposition p_1 given by a user, and he wants to know the truth value of proposition p_4. The lFRSN P system model of these fuzzy production rules is constructed, as shown in Figure 7, to find the truth value of proposition p_4 which is the potential value of σ_4.

From Figure 7, there are three paths from neuron σ_1 to neuron σ_4 and are described as follows:

path (1) : $\sigma_1 \to \sigma_8 \to \sigma_7 \to \sigma_{11} \to \sigma_4$
path (2) : $\sigma_1 \to \sigma_9 \to \sigma_2 \to \sigma_{12} \to \sigma_4$
path (3) : $\sigma_1 \to \sigma_{10} \to \sigma_6 \to \sigma_{14} \to \sigma_4$

According to computing mechanism of neurons in lFRSN P systems, we get results described as follows: from path (1), potential value of σ_4 is: *very true* $\otimes$ *almost certain* $\otimes$ *sort of true* $=$ *sort of true*; from path (2), potential value of σ_4 is: *very true* $\otimes$ *almost certain* $\otimes$ *rather true* $=$ *rather true*; from path (3), potential value of σ_4 is: *very true* $\otimes$ *pretty true* $\otimes$ *very true* $=$ *pretty true*. Therefore, potential value of σ_4 is: *sort of true* $\oplus$ *rather true* $\oplus$ *pretty true* $=$
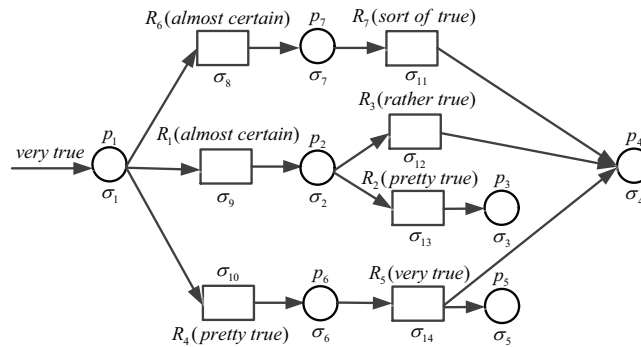
Figure 7: An lFRSN P system model for seven fuzzy production rules.

pretty true. Thus, truth value of proposition is *pretty true.*

3.2 Transformers

In this subsection, an application example is used to demonstrate the effectiveness of rFRSN P systems and their FRA in fault diagnosis of a transformer. The following fuzzy production rules are obtained from the knowledge base of a transformer fault diagnosis system [12].

Rule 1 (CF=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 (CF=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 (CF=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 (CF=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 rFRSN P system Π_5, as shown in Figure 8.

$\Pi_5 = (A, \sigma_1, \ldots, \sigma_{14}, \sigma_{15}, \ldots, \sigma_{18}, syn, I, O)$, 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 parameter vectors and matrices given in [12], U, V, Λ, H_1 and H_2 are follows:

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

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

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

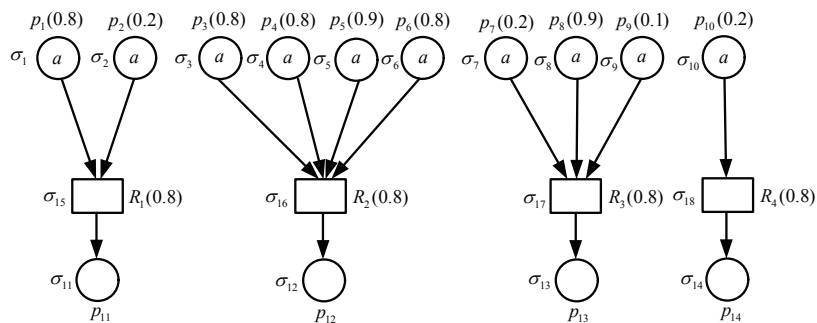$$\lambda_p = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)^T \qquad \lambda_r = (2, 4, 4, 3)^T$$

Figure 8: An example of a transformer fault diagnosis modeled by an rFRSN P system model Π_5.

In on-scene information detection of transformer, total hydrocarbon content is high (CF=0.8), C_2H_2 content is high (CF=0.8), H_2 content is high (CF=0.9), C_2H_2 content in total hydrocarbon content is little (CF=0.8), CH_4 content in total hydrocarbon content is little (CF=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 *subsection 2.4*, we get

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

(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) $a_p^3 = (0,0,0,0,0,0,0,0,0,0,0,0,0,0)^T$, $a_r^3 = (0,0,0,0)^T$.

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 (CF=0.16), serious overheating fault (CF=0.64), partial discharge (CF=0.08) and spark discharge (CF=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.

3.3   Traction power supply systems

In [19], three cases from the local system of a TPSS chosen in [49], as shown in Figure 9, are considered as examples to manifest the effectiveness of WFRSN P systems in fault diagnosis, where the external transmission lines in a power system which supplies the TPSS are hypothetical, S and R represent the sending end and receiving end of transmission lines, L represents transmission lines. The first two cases are in normal power supply and the third case is in over zone feeding. It is worth pointing out that, the complete line connection of FS1, ATP1, SP1, FS3, ATP3 and TPS-02 is the same as that of TSS-01, FS2, SP2 and ATP2 in Figure 9.
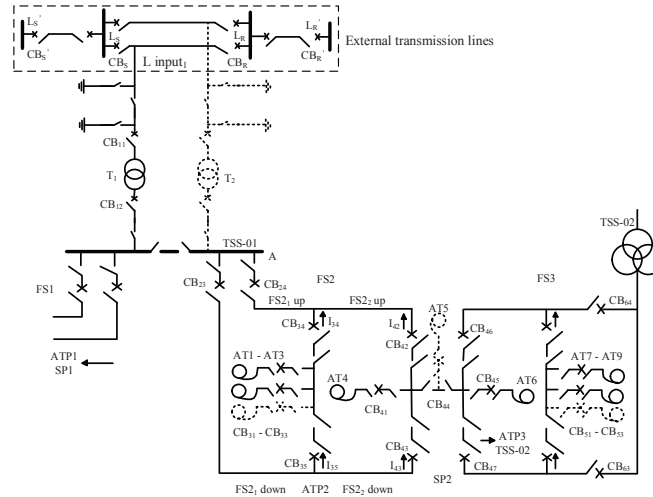

Figure 9: A local single line sketch map of a TPSS.

Case 1: normal power supply. $FS2_{1\ up}$ and AT1 have faults.

Status information from the SCADA system (in time order): $AT1_m$ operated, CB_{31} tripped, $AT3$ auto switched over; $FS2_m$ operated, CB_{23} and CB_{24} tripped; feeder lines auto reclosed, $FS2_{up\ m}$ operated quickly, CB_{23} tripped again. When faults occur, current directions of I_{34} and I_{35} are positive, and current is not detected in SP2.

A WFRSN P system for $FS2_{up}$ is Π_6 and its corresponding WFRSN P system fault diagnosis model is shown in Figure 10.

$\Pi_6 = (O, \sigma_1, \ldots, \sigma_{16}, syn, in, out)$, where:

(1) $O = \{a\}$ is the singleton alphabet (a is called spike);

(2) $\sigma_1, \ldots, \sigma_9$ are proposition neurons corresponding to the propositions with truth values $\theta_1, \ldots, \theta_9$; that is, $s = 9$;

(3) $\sigma_{10}, \ldots, \sigma_{13}$ are rule neurons, where σ_{10}, σ_{11} and σ_{12} are *and* rule neurons, σ_{14} is an *or* rule neuron; that is, $t = 4$;

(4) $syn = \{(1, 10), (2, 10), (2, 11), (3, 11), (4, 12), (5, 12), (6, 13), (7, 13), (8, 13), (10, 6), (11, 7), (12, 8), (13, 9)\}$;

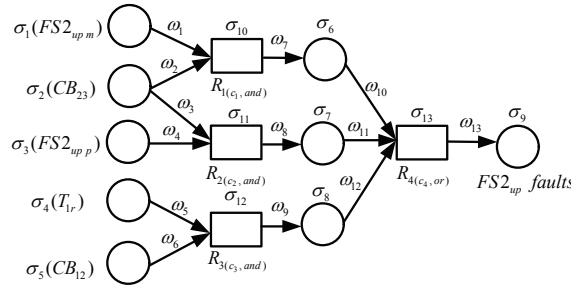(5) $in = \{\sigma_1, \ldots, \sigma_5\}$, $out = \{\sigma_9\}$.

Figure 10: A WFRSN P system fault diagnosis model for $FS2_{up}$.

The synaptic weight matrices of Π_2 are shown in Figure 11 and other parameter matrices associated with the model in Figure 10 are described as follows: $\boldsymbol{\theta_0} = (0.9913\ 0.9833\ 0.8\ 0.4\ 0.2\ 0\ 0\ 0\ 0)^T$, $\boldsymbol{\delta_0} = (0\ 0\ 0\ 0)^T$, $\boldsymbol{C} = diag(0.975\ 0.95\ 0.9\ 0.975)$. In order to succinctly describe the matrices, let us denote $\boldsymbol{O_l} = (x_1, \ldots, x_l)^T$, where $x_i = 0, 1 \leq i \leq l$. When $g = 0$, we get the results: $\boldsymbol{\delta_1} = (0.9873\ 0.8917\ 0.3\ 0)^T$, $\boldsymbol{\theta_1} = (0\ 0\ 0\ 0\ 0\ 0\ 0.9626\ 0.8471\ 0.27\ 0)^T$. When $g = 1$, we get the results: $\boldsymbol{\delta_2} = (0\ 0\ 0\ 0.9626)^T$, $\boldsymbol{\theta_2} = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0.9385)^T$. When $g = 2$, we get the results: $\boldsymbol{\delta_3} = (0\ 0\ 0\ 0)^T$. Thus, the termination condition is satisfied and the reasoning process ends. We obtain the reasoning results, i.e., the truth value 0.9385 of the output neuron σ_9. The feeding section $FS2_{up}$ has a fault with a fault confidence level 0.9385. The fault region of $FS2_{up}$ can be further identified according to the fault current detected and the WFRSN P system fault diagnosis model for fault region identification in Figure 12, and then we get the result that $FS2_{1\ up}$ has a fault with a fault confidence level 0.9385.

For AT1, a WFRSN P system can be constructed in a similar way and its corresponding WFRSN P system fault diagnosis model is shown in Figure 13. The diagnosis process of AT1 is similar. According to the SCADA data and Table 3, the parameter matrices of WFRSN P system fault diagnosis model for AT1 is established to perform WMBRA. After the reasoning, the fault confidence level of AT1 is obtained, i.e., 0.8361. So the autotransformer AT1 has a fault with a fault confidence level 0.8361.

Case 2: normal power supply. $FS2_{1\ up}$ has faults.

$$
\boldsymbol{W_{r1}} = [\mathbf{O}]_{9 \times 4}, \boldsymbol{W_{r2}} = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \boldsymbol{W_{r3}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \boldsymbol{W_p} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.
$$

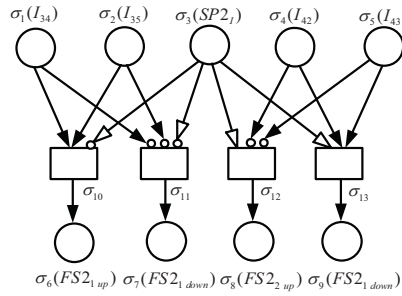Figure 11: Synaptic weight matrices of WFRSN P system fault diagnosis model for $FS2_{up}$.

Figure 12: A WFRSN P system fault diagnosis model for fault region identification of a feeding section.
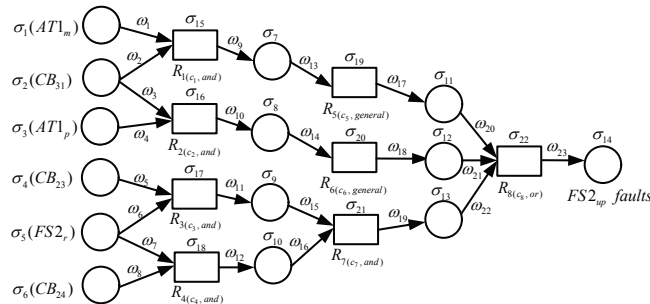

Figure 13: A WFRSN P system fault diagnosis model for AT1.

Sections	Protective devices (operated)						Protective devices (non-operated)					
	Main		Primary backup		Remote backup		Main		Primary backup		Remote backup	
	Relays	CBs	Relays	CBs	Relays	CBs	Relays	CBs	Relays	CBs	Relays	CBs
FL	0.9913	0.9833	0.8	0.85	0.7	0.75	0.2	0.2	0.2	0.2	0.2	0.2
B	0.8564	0.9833	-	-	0.7	0.75	0.4	0.2	-	-	0.4	0.2
T	0.7756	0.9833	0.75	0.8	0.7	0.75	0.4	0.2	0.4	0.2	0.4	0.2

Table 3: Operation and non-operation confidence levels of the protective devices

Status information from the SCADA system (in time order): $FS2_m$ operated, CB_{24} tripped; T_{1r} operated, CB_{11} and CB_{12} tripped. When faults occur, current directions of I_{34} and I_{35} are positive, and current is not detected in SP2. In this case, CB_{23} refused operation.

According to the SCADA data and Table 3, the WFRSN P system fault diagnosis model for

$FS2_1$ and its parameter matrices are established to perform WMBRA. After the reasoning, the fault confidence level of $FS2_{up}$ is obtained, i.e., 0.7439. The fault region of $FS2_{up}$ can be further identified according to the fault current detected and the WFRSN P system fault diagnosis model for fault region identification in Figure 12, and then we get the result that $FS2_{1\ up}$ has a fault. So the feeding section $FS2_{1\ up}$ has a fault with a fault confidence level 0.7439.

Case 3: FS2 is over zone fed by TPS-02. AT7 and $FS2_{2\ up}$ have faults.

Status information from the SCADA system (in time order): primary backup protections of feeder lines in SP2 operated, CB_{42} tripped; meanwhile, CB_{51} tripped, AT9 auto switched over; remote backup protection $FS3_s$ of feeder lines in TSS-02 operated, CB_{63} and CB_{64} tripped. When faults occur, current directions of I_{34} and I_{35} are positive, and current is detected only in SP2 and ATP2. In this case, main protection of feeder lines in SP2, CB_{43} and main protection of AT7 refused operation, and status information of primary backup protection of AT7 lost.

According to the SCADA data and Table 3, the WFRSN P system fault diagnosis models for AT7 and $FS2_2$ and their parameter matrices are established to perform WMBRA, respectively. After the reasoning, the fault confidence levels AT7 and $FS2_{up}$ are obtained, i.e., 0.6946 and 0.6123. The fault region of $FS2_{up}$ can be further identified according to the fault current detected and the WFRSN P system fault diagnosis model for fault region identification in Figure 12, and then we get the result that $FS2_{2\ up}$ has a fault. So the autotransformer AT2 has a fault with a fault confidence level 0.6946 and the feeding section $FS2_{2\ up}$ has a fault with a fault confidence level 0.6123.

The results of *Cases 1-3* give evidence of that the proposed fault diagnosis approach can obtain satisfying results both in the situation in normal power supply and over zone feeding with complete/incomplete alarm information. In addition, the proposed method can provide results comparable with those in [49] by using only one simple reasoning while the method in [49] needs a second reasoning.

3.4   Power transmission networks

AFRSN P systems

In this section, four cases of the local system in an EPS shown in Figure 14 are considered as examples to show the effectiveness and superiority of AFRSN P systems for fault diagnosis of power transmission networks. These cases include a single and multiple fault situations [21]. The status information (with/without incompleteness and uncertainty) about protective relays and CBs, and diagnosis results based on AFRSN P systems are shown in Table 4, where "$*$" means that this case includes incomplete or uncertain status information from the SCADA system. According to Table 4, we know that the fault diagnosis models based on AFSN P systems can deal with the uncertainty of action messages about protective relays and breakers. Furthermore, fault element is diagnosed properly while the information is incomplete because of the well fault tolerance. Therefore, this method is effective in fault diagnosis.

tFRSN P systems

Fault diagnosis method based on tFRSN P systems is called FDSNP for short [25]. This subsection summarizes the FDSNP method, whose flowchart is shown in Figure 15, as follows.

Step 1 Read operation messages about protective relays and/or CBs in a power transmission network from the SCADA system.

Step 2 Search for outage areas. We suggest network topology analysis because it decreases the number of candidate diagnosing areas and reduce the subsequent computational workload [50]. The search process is described as follows:
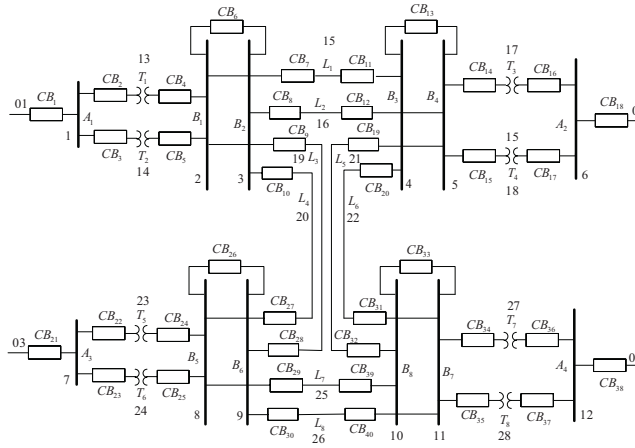
Figure 14: A local sketch map of the protection system of an EPS.

Cases	Status information		Diagnosis results	
	Operated relays	Tripped CBs	Fault section	Fuzzy truth value
1	B_{1m}	$CB_4, CB_5, CB_7, C9, CB_{12}, CB_{27}$	B_1	0.87386
2	B_{1m}, L_{2Rs}, L_{4Rs}	$CB_4, CB_5, CB_7, C9, CB_{12}, CB_{27}$	B_1	0.78044
3	B_{1m}, L_{1Sp}, L_{1Rm}	$CB_4, CB_5, CB_6, CB_7, CB_9, CB_{11}$	B_1, L_1	0.87386, 0.86085
4^*	L_{2Rs}, L_{4Rs}	$CB_4, CB_5, CB_7, CB_9, CB_{12}, CB_{27}$	B_1	0.56363

Table 4: Status information and diagnosis results based on AFRSN P systems

(i) Let the search iteration $t = 1$;

(ii) Construct a set Q_t of section numbers: assign a number to each section in the power transmission network. The numbers of all sections constitute the set Q_t;

(iii) Construct a subset M_t of section numbers: put the number of a randomly chosen section from Q_t into the subset M_t. If there is a closed CB connecting this chosen section, find all the closed CBs connecting it, otherwise, go to *(vi)*. Find all the other sections linking with each of the closed CBs and put their numbers from Q_t into M_t. Continue to find the closed CBs and sections according to those in M_t;

(iv) t is increased by one;

(v) Construct the set Q_t: remove the numbers of the sections in M_t from Q_{t-1} and obtain Q_t. If Q_t is not empty, the search process goes to *(iii)*;

(vi) Find passive networks, i.e., outage areas, from $M_1, M_2, \ldots, M_{N_s}$, where N_s is the maximum number of all numbers referring to section subsets. The search process stops.

Step 3 If there is only one section in the passive networks found in *Step 2*, this section is the faulty one and the algorithm stops, otherwise, a fault diagnosis model based on an tFRSN P system is built for each section. The model-building process is described as follows. A section in the passive network is chosen randomly. According to the relay protections of the section, we design fault fuzzy production rules and then determine proposition and rule neurons and create their linking relationship to obtain the tFRSN P system. The certainty factor of each rule is empirically set. According to Tables 5 and 6, we set confidence levels for main protections, first backup protections, second backup protections and their CBs. Then a one-to-one relationship between the fuzzy truth value of each input neuron and the confidence level of each protection is established to obtain the initial values of the model.

Sections	Protective devices					
	Main		First backup		Second backup	
	Relays	CBs	Relays	CBs	Relays	CBs
L	VH	VH	H	H	MH	MH
B	VH	VH	-	-	MH	MH
T	VH	VH	H	H	MH	MH

Table 5: Confidence levels of the operated protective devices

Sections	Protective devices					
	Main		First backup		Second backup	
	Relays	CBs	Relays	CBs	Relays	CBs
L	L	L	L	L	L	L
B	ML	L	-	-	ML	L
T	ML	L	ML	L	L	L

Table 6: Confidence levels of the non-operate protective devices

Step 4 The algebraic fuzzy reasoning algorithm is used to acquire the fault confidence level of each section.

Step 5 If the confidence level θ of a section satisfies the condition $\theta \geq (0.58, 0.63, 0.80, 0.86)$, the section is faulty, otherwise, if θ satisfies the condition $\theta \leq (0.17, 0.22, 0.36, 0.42)$, the section is not faulty, otherwise, the section may be faulty.
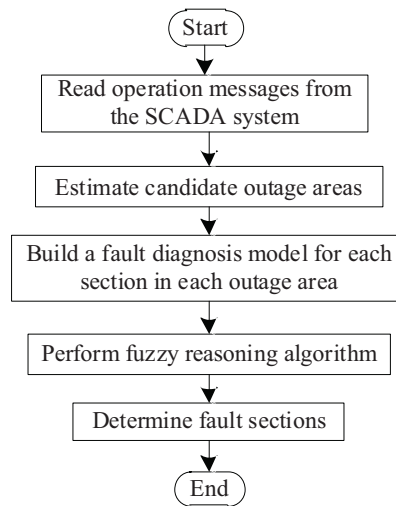

Figure 15: The flowchart of FDSNP.

In [25], seven cases of the local system in an EPS shown in Figure 14 are considered as examples to test the effectiveness and superiority of FDSNP. These cases include single and multiple fault situations. The status information (with/without incompleteness and uncertainty) about protective relays and CBs is shown in Table 7, where "$*$" means that this case includes incomplete or uncertain status information from the SCADA system.

FDSNP is used to diagnose faults for the seven cases, and the diagnosis results are shown in Table 8, which contains the faulty sections and their fault confidence levels. Table 8 lists the

Cases	Status information	
	Operated relays	Tripped CBs
1	B_{1m}, L_{2Rs}, L_{4Rs}	$CB_4, CB_5, CB_7, CB_9, CB_{12}, CB_{27}$
2*	L_{2Rs}, L_{4Rs}	$CB_4, CB_5, CB_7, CB_9, CB_{12}, CB_{27}$
3	B_{1m}, L_{1Sp}, L_{1Rm}	$CB_4, CB_5, CB_6, CB_7, CB_9, CB_{11}$
4	$B_{1m}, L_{1Sm}, L_{1Rp}, B_{2m}, L_{2Sp}, L_{2Rm}$	$CB_4, CB_5, CB_6, CB_7, CB_8, CB_9, CB_{10}, CB_{11}, CB_{12}$
5	T_{3p}, L_{7Sp}, L_{7Rp}	$CB_{14}, CB_{16}, CB_{29}, CB_{39}$
6	$L_{1Sm}, L_{1Rp}, L_{2Sp}, L_{2Rp}$	$CB_7, CB_8, CB_{11}, CB_{12}$
	$L_{7Sp}, L_{7Rm}, L_{8Sm}, L_{8Rm}$	$CB_{29}, CB_{30}, CB_{39}, CB_{40}$
7*	$T_{7m}, T_{8P}, B_{7m}, B_{8m}, L_{5Sm}$	$CB_{19}, CB_{20}, CB_{29}, CB_{30}, CB_{32}$
	$L_{5Rp}, L_{6Ss}, L_{7Sp}, L_{7Rm}, L_{8Ss}$	$CB_{33}, CB_{34}, CB_{35}, CB_{36}, CB_{37}, CB_{39}$

Table 7: Status information about protective relays and CBs

Cases	Diagnosis results of FDSNP		
	Fault sections	Fault confidence levels	
		Trapezoidal fuzzy numbers	Linguistic terms
1	B_1	(0.975, 0.98, 1, 1)	VH
2	B_1	(0.5655, 0.6174, 0.80, 0.86)	[M, MH]
3	B_1, L_1	(0.975, 0.98, 1, 1), (0.9506, 0.9604, 1, 1)	VH, [H, VH]
4	B_1	(0.975, 0.98, 1, 1)	VH
	B_2	(0.975, 0.98, 1, 1)	VH
	L_1	(0.9506, 0.9604, 1, 1)	[H, VH]
	L_2	(0.9506, 0.9604, 1, 1)	[H, VH]
5	T_3	(0.72, 0.78, 0.92, 0.97)	H
	L_7	(0.9506, 0.9604, 1, 1)	[H, VH]
6	L_1	(0.702, 0.7644, 0.92, 0.97)	[H, VH]
	L_2	(0.702, 0.7644, 0.92, 0.97)	[H, VH]
	L_7	(0.702, 0.7644, 0.92, 0.97)	[H, VH]
	L_8	(0.9506, 0.9604, 1, 1)	[H, VH]
7	L_5	(0.702, 0.7644, 0.92, 0.97)	[H, VH]
	L_7	(0.702, 0.7644, 0.92, 0.97)	[H, VH]
	B_7	(0.975, 0.98, 1, 1)	[H, VH]
	B_8	(0.975, 0.98, 1, 1)	[H, VH]
	T_7	(0.975, 0.98, 1, 1)	[H, VH]
	T_8	(0.72, 0.78, 0.92, 0.97)	H

Table 8: Fault sections and their fault confidence levels obtained by using FDSNP

fault confidence levels of only faulty sections. Four diagnosis methods, fuzzy logic (FL) [51], fuzzy Petri nets (FPN) [32], genetic algorithm-tabu search (GATS) [52] and genetic algorithm (GA) [53], are used as benchmarks to perform comparative experiments. The diagnosis results of the five methods are shown in Table 9, where " $-$ " means that this case was not considered in the corresponding reference.

From Table 8, we can see that the fault confidence levels represented by trapezoidal fuzzy numbers provide a quantitative description for the faulty sections which makes these results

Cases	Diagnosis results				
	FDSNP	FL [51]	FPN [32]	GATS [52]	GA [53]
1	B_1	B_1	B_1	-	B_1
2	B_1	-	-	-	-
3	B_1, L_1	B_1, L_1	B_1, L_1	-	B_1, L_1
4	B_1, B_2, L_1, L_2	B_1, B_2, L_1, L_2	B_1, B_2, L_1, L_2	-	B_1, B_2, L_1, L_2
5	T_3, T_7	T_3, T_7	T_3, T_7	T_3, T_7	$(1)T_3, L_7; (2)T_3$ $(3)L_7; (4)No$
6	L_1, L_2 L_7, L_8	L_1, L_2 L_7, L_8	L_1, L_2 L_7, L_8	L_1, L_2 L_7, L_8	$(1)L_1, L_2, L_7, L_8$ $(2)L_1, L_7, L_8$
7	L_5, L_7, B_7 B_8, T_7, T_8	L_5, L_7, B_8 T_7, T_8	L_5, L_7, B_7 B_8, T_7, T_8, L_8	L_5, L_7, B_7 B_8, T_7, T_8	$(1)L_5, L_7, B_7, B_8, T_7, T_8$ $(2)L_5, L_7, T_7, B_8$

Table 9: Comparisons between FDSNP and four fault diagnosis methods

more reliable. The linguistic terms corresponding to these trapezoidal fuzzy numbers provide a more intuitive and flexible way for experts and dispatchers than probability values, since their knowledge usually contain linguistic terms with a certain degree of uncertainty.

From Table 9, we can see that the diagnosis results of FDSNP, in *Case 1* and *Cases 3-6*, are the same as those in [32, 51], in other words, FDSNP is effective in fault diagnosis of power transmission networks in power systems. Table 9 also shows that, in some cases, FDSNP is superior to FL, FPN and GA on correctly identifying fault sections. For instance, in *Case 7*, the fault diagnosis result of FDSNP is different from those in [32, 51, 53]. In this case, for section L_8, only its second backup protective relay S_{L8Ss} operated and actually, S_{L8Ss} operated as the second backup protective relay of section B_8. So in fact, L_8 is not a faulty section. For section B_7, its main protective relay B_{7m} operated and tripped its corresponding CBs, CB_{33}, CB_{34} and CB_{35}. So B_7 is a faulty section. Thus, for *Case 7*, the diagnosis result of FDSNP and GATS is better than those in [32, 51, 53]. In *Cases 5-7*, comparisons of diagnosis results between FDSNP and the methods in [53] show that FDSNP can solve the nonuniqueness problem of the diagnosis solution, which proves the correctness of FDSNP in diagnosing faulty sections. Besides, the diagnosis results in *Cases 2* and *7* show that FDSNP can obtain satisfying results in the situations with incomplete or uncertain alarm information. Therefore, from the seven typical cases, FDSNP is effective with a good accuracy in fault diagnosis of power transmission networks.

4 Conclusions and future research lines

Fuzzy membrane computing is an important research branch of membrane computing. Until now there have been two main fuzzy P systems: fuzzy cell-like P systems and fuzzy reasoning spiking neural P systems. In this survey, the theoretical developments and applications about fuzzy membrane computing are summarized. To advance this research direction, we list some promising topics as follows:

(1) Extensions and applications of fuzzy cell-like P systems. From the survey in Sections 2.1 and 2.2 one can see that only a limited work about considering fuzzy approaches in the framework of cell-like P systems, but a broad variety of fuzzy cell-like P systems can be further investigated.

- For simplicity, only symport/antiport P systems whose rules only move reactants through

membranes were considered to deal with uncertainty in [43], but it is straightforward that this fuzzy approach can be extended to other variants of cell-like P systems.

- Fuzzy mathematics have been used to handle the uncertainty in the number of copies of the reactants, imperfectness of objects in membranes and approximate copies of reactants involved in reactions. These fuzzy mathematics now involve t-norm approach (Triangular norms), fuzzy sets, (general) multi-fuzzy sets, L-multi fuzzy sets, L-fuzzy hybrid sets and so on. Other fuzzy mathematics can also be considered.

- Feasibility of applying fuzzy cell-like P systems in computational biology was discussed in [44], but until now no real application case appears in research literature. So the real application of fuzzy cell-like P systems in computational biology and other fields is a promising research line.

(2) Introducing fuzzy approach into tissue-like P systems. Uncertainty in an inherent property in all living systems and tissue-like P systems are models inspired by the way the tissue cells lives and functions. So uncertainty also should be considered in tissue-like P systems due to the real behavior of tissue cells.

(3) Fault diagnosis method based on FRSN P systems (FDM-FRSNP, for short) is summarized in this paper. However, up to now, the work focuses on the effectiveness and correctness of FDM-FRSNP and the results of application examples are obtained by manual computation.

-To test the speed, convergence and accuracy of the fuzzy reasoning algorithms of FRSN P systems, and to explore automatical generation of FRSN P systems in fault diagnosis, our future work will simulate them on MATLAB [54], P-Lingua [55] or MeCoSim [56]. Meanwhile, how to verify and realize the parallelism of FRSN P systems and their fuzzy reasoning algorithm on hardware such as FPGA and CUDA is also our further task.

-Valuable research interests refer to extend models, algorithms and application areas. For models and algorithms, one promising topic is to design new variants of SN P systems and their reasoning algorithms according to requirements of different fault diagnosis problems, such as on-line diagnosis, fast fault diagnosis, high-precision diagnosis. Another promising interest is to propose FRSN P systems with learning ability. For application areas, FRSN P systems can be used to more different systems, such as power supply systems for urban rail transit, mechanical fault diagnosis and power systems with new energies.

Acknowledgment

Bibliography

[1] Păun, G. (2000); Computing with Membranes, *J Comput. Syst. Sci.*, ISSN 0022-0000, 61(1): 108–143.

[2] Păun, G.; Rozenberg, G.; Salomaa, A. (2010); The Oxford Handbook of Membrane Computing, Oxford University Press, New York, NY, USA.

[3] Zhang, G.X.; Gheorghe, M.; Pan L.Q.; Pérez-Jiménez, M.J. (2014); Evolutionary Membrane Computing: A Comprehensive Survey and New Results, *Inform. Sciences*, ISSN 0020-0255, 279: 528-551.

[4] Ionescu, M.; Păun, Gh.; Yokomori T. (2006); Spiking Neural P Systems, *Fund. Inform.*, ISSN 0169-2968, 71(2-3): 279–308.

[5] Păun, G.; Pérez-Jiménez, M.J.; Rozenberg, G. (2006); Spike Trainss in Spiking Neural P Systems, *Int. J. Found. Comput. S.*, ISSN 0129-0541, 17(4): 975-1002.

[6] Cavaliere, M.; Ibarra, O.H.; Păun, Gh.; Egecioglu, O.; Ionescu, M.; Woodworth, S. (2009); Asynchronous Spiking Neural P Systems, *Theor. Comput. Sci.*, ISSN 0304-3975, 410(24-25): 2352-2364.

[7] Pan, L.Q.; Zeng, X.X. (2011); Small Universal Spiking Neural P Systems Working in Exhaustive Mode, *IEEE T. Nanobiosci.*, ISSN 1536-1241, 10(2): 99-105.

[8] Zhang, X.Y.; Luo, B.; Fang, X.Y.; Pan, L.Q. (2012); Sequential spiking neural P systems with exhaustive use of rules, *BioSystems*, ISSN 0303-2647, 108(1-3): 52-62.

[9] Francis G.C.; Henry N.A. (2012); On Structures and Behaviors of Spiking Neural P Systems and Petri Nets, *Lect. Notes in Comput. Sc.*, ISSN 0302-9743, vol. 7762: 145-160.

[10] Song, T.; Pan, L.Q.; Păun, G. (2013); Asynchronous Spiking Neural P Systems with Local Synchronization, *Inform. Sciences*, ISSN 0020-0255, 219: 197-207.

[11] Zhang, G.X., Rong H.N., Neri, F., Pérez-Jiménez M.J. (2014); An Optimization Spiking Neural P System for Approximately Solving Combinatorial Optimization Problems, *Int. J. Neural. Syst.*, ISSN: 0129-0657, 24(5): 1440006 (16 pages).

[12] Peng, H.; Wang, J.; Pérez-Jiménez, M.J.; Wang, H.; Shao, J.; Wang, T. (2013); Fuzzy Reasoning Spiking Neural P System for Fault Diagnosis, *Inform. Sciences*, ISSN 0020-0255, 235(20): 106-116.

[13] Xiong, G.J.; Shi, D.Y.; Chen, J.F. (2013); Implementing fuzzy reasoning spiking neural P system for fault diagnosis of power systems, IEEE Power Energy Soc. Gen. Meet., Article ID 5970635, 5 pages.

[14] Xiong, G.J.; Shi, D.Y.; Zhu, L.; Duan, X.Z. (2013); A New Approach to Fault Diagnosis of Power Systems Using Fuzzy Reasoning Spiking Neural P Systems, *Math. Probl. Eng.*, ISSN 1024-123X, vol. 2013: Article ID 815352, 13 pages.

[15] Wang T.; Zhang G.X.; Pérez-Jiménez M.J. (2014); Fault Diagnosis Models for Electric Locomotive Systems Based on Fuzzy Reasoning Spiking Neural P Systems, *Lect. Notes Comput. Sci. (CMC 2014)*, vol. 8961, pp. 361-374.

[16] Wang, J.; Peng, H. (2010); Fuzzy Knowledge Representation Based on An Improving Spiking Neural P Systems, *Proc. ICNC*, pp. 3012-3015.

[17] Wang, J.; Peng, H. (2011); An Extended Spiking Neural P Systems for Fuzzy Knowledge Representation, *Int. J. Innov. Comput. Inf. Control*, ISSN 1349-4198, 7(7): 3709-3724.

[18] Wang, J.; Shi, P.; Peng, H.; Pérez-Jiménez M.J.; Wang, T. (2013); Weighted Fuzzy Spiking Neural P Systems, *IEEE T. Fuzzy Syst.*, ISSN 1063-6706, 21(2): 209-220.

[19] Wang T.; Zhang G.X.; Pérez-Jiménez M.J.; Cheng J.X. (2015); Weighted Fuzzy Reasoning Spiking Neural P Systems: Application to Fault Diagnosis in Traction Power Supply Systems of High-Speed Railways, *J. Comput. Theor. Nanos.*, ISSN 1546-1955, 12(7): 1103-1114.

[20] Wang, J.; Peng, H. (2013); Adaptive Fuzzy Spiking Neural P Systems for Fuzzy Inference and Learning, *Int. J. Comput. Math.*, ISSN 0020-7160, 90(4): 857-868.

[21] Tu M.; Wang, J.; Peng, H.; Shi Peng (2014); Application of Adaptive Fuzzy Spiking Neural P Systems in Fault Diagnosis of Power Systems, *Chinese J. Electron.*, ISSN 1022-4653, 23(1): 87-92.

[22] Wang, T.; Wang, J.; Peng, H.; Deng Y.L. (2010); Knowledge Representation Using Fuzzy Spiking Neural P systems, *Proc. IEEE BIC-TA*, pp. 586-590.

[23] Wang, T.; Wang, J.; Peng, H.; Wang, H. (2011); Knowledge Representation and Reasoning Based on FRSN P System, *Proc WCICA*, pp. 849-854.

[24] Wang T.; Zhang G.X.; Rong H.N.; Pérez-Jiménez M.J. (2014); Application of Fuzzy Reasoning Spiking Neural P Systems to Fault Diagnosis, *Int. J. Comput. Commun.*, ISSN 1841-9836, 9(6): 786-799.

[25] Wang, T.; Zhang, G.X.; Zhao, J.B; He, Z.Y; Wang, J.; Pérez-Jiménez M.J. (2015); Fault Diagnosis of Electric Power Systems Based on Fuzzy Reasoning Spiking Neural P Systems, *IEEE T. Power Syst.*, ISSN 0885-8950, 30(3): 1182-1194.

[26] Zadeh, L.A. (1965); Fuzzy Sets, *Inform. Contr.*, ISSN: 0019-9958, 8(3): 328-353.

[27] Guiffrida, A.L.; Nagi, R. (1998); Fuzzy Set Theory Applications in Production Management Research: A Literature Survey, *J. Intell. Manuf.*, ISSN: 0956-5515, 9(1): 39-56.

[28] Zadeh, L.A. (1994); Fuzzy Logic Technology and Their Application, *IEEE Publications*, 1994.

[29] Chen, S.M. (1996); A Fuzzy Reasoning Aproach for Rule-based Systems Based on Fuzzy Logics, *IEEE T. Syst. Man Cy. B*, ISSN 1083-4419, 26(5): 769-778.

[30] Liu, H.C.; Liu, L.; Lin, Q.L.; Liu, N. (2013); Knowledge Acquisition and Representation Using Fuzzy Evidential Reasoning and Dynamic Adaptive Fuzzy Petri Nets, *IEEE T. Cybern.*, ISSN 1083-4419, 43(3): 1059-1072.

[31] Chen, W.H. (2011); Fault Section Estimation Using Fuzzy Matrix-based Reasoning Methods,*IEEE T. Power Deliver.*, ISSN 0885-8977, 26(1): 205-213.

[32] Sun, J.; Qin, S.Y.; Song, Y.H. (2004); Fault Diagnosis of Electric Power Systems Based on Fuzzy Petri Nets, *IEEE T. Power Syst.*, ISSN 0885-8950, 19(4): 2053-2059.

[33] Choi, C.; Kim, C.; Sung, N.; Park, Y. (2007); Evaluating the Quality of Service in Mobile Business Based on Fuzzy Set Theory, *Proc. FSKD*, pp. 483-487.

[34] Chaira, T. (2012); Medical Image Enhancement Using Intuitionistic Fuzzy Set, *Proc. RAIT*, pp. 54-57.

[35] Păun, Gh. (2000); Computing with Membranes (P Systems): Twenty Six Research Topics, available at http://psystems.disco.unimib.it/download/probl.pdf.

[36] Nola, A.D.; Păun, G.; Pérez-Jiménez, M.J.; Rosselló, F. (2004); (Imprecise Topics about) Handling Imprecision in P Systems, *Proc. BWUMC*, pp. 1–10.

[37] Păun, G. (2005); Further Twenty Six Open Problems in Membrane Computing, *Proc. BWMC*, pp. 249-262.

[38] Păun, G. (2007); Tracing Some Open Problems in Membrane Computing, *Proc. BWMC*, ISSNl 1453-8245, 10(4): 303-314.

[39] Obtułowicz A.; Păun, Gh. (2003); (In Search of) Probabilistic P Systems, *Biosystems*, ISSNl 0303-2647, 70(2): 107-121.

[40] Obtułowicz A. (2003); Mathematical Models of Uncertainty with A Regard to Membrane Systems, *Nat. Comput.*, ISSN 1567-7818, 2: 251–263.

[41] Obtułowicz A. (2005); General Multi-fuzzy Sets and Fuzzy Membrane Systems, *Lect. Notes Comput. Sci.*, ISSN 0302-9743, 3365: 359-372.

[42] Casasnovas, J.; Miro, J.; Moyá, M.; Rosselló, F. (2004); A Fuzzy Approach to Membrane Computing with Approximate Copies, *Proc. BWUMC*: 121–127.

[43] Casasnovas, J.; Miro, J.; Moyá, M.; Rosselló, F. (2004); An Approach to Membrane Computing Under Inexactitude, *Int. J. Found. Comput. S.*, ISSN 0129-0541, 15(6): 841-864.

[44] Casasnovas, J.; Rosselló, F. (2005); Fuzzy P Systems and Their Applications in Computational Biology, *Proc. EUSFLAT*, pp. 1112-1117.

[45] Aguzzoli, S.; Besozzi, D., Gerla, B., Manara, C. (2004); P Systems with Vague Boundaries: the T-norm Approach, *Proc. BWUMC*, pp. 97–105.

[46] Aguzzoli, S.; Ardelean, I.I.; Besozzi, D.; Gerla, B., Manara, C (2004); P Systems Under Uncertainty: the Case of Transmembrane Proteins, *Proc. BWUMC*, pp. 107–117.

[47] Syropoulos A. (2006); Fuzzifying P Systems, *Comput. J.*, ISSN 0010-4620, 49(5): 619-628.

[48] Syropoulos, A. (2012); On Generalized Fuzzy Multisets and Their Use in Computation, *Iran. J. Fuzzy Syst.*, ISSN 1735-0654, 9(2): 113-125.

[49] Wu, S.; He, Z.Y.; Qian, C.H.; Zang, T.L. (2011); Application of Fuzzy Petri Net in Fault Diagnosis of Tranction Power Supply System for High-speed Way, *Power System Tech.*, ISSN 1000-3673, 35(9): 79-85.

[50] Yang, J.W.; He, Z.Y.; Zang, T.L. (2010); Power System Fault-diagnosis Method Based on Directional Weighted Fuzzy Petri Nets, *Proc. CSEE.*, ISSN 0258-8013, 30(34): 42-49.

[51] Chang, C.S.; Chen, J.M.;Srinivasan, D., Wen, F.S.; Liew, A.C. (1997); Fuzzy Logic Approach in Power System Fault Section Identification, *IEE P. Gener. Transm. D.*, 144(5): 406-414.

[52] Lin, X.N.; Ke, S.H.; Li, Z.T.; Weng, H.L.; Han, X.H. (2010); A Fault Diagnosis Method of Power Systems Based on Improved Objective Function and Genetic Algorithm-tabu Search, IEEE T. Power Deliver., ISSN 0885-8977, 25(3): 1268-1274.

[53] Wen, F.S.; Han, Z.X. (1995); Fault Section Estimation in Power Systems Using a Genetic Algorithm," *Electr. Pow. Syst.s Res.*, ISSN 0378-7796, 34(3): 165-172.

[54] The Matlab Website. http://www.mathworks.es/products/matlab/.

[55] Research Group on Natural Computing, University of Seville: The P-Lingua Website. http://www.p-lingua.org.

[56] Research Group on Natural Computing, University of Seville: The MeCoSim Website. http://www.p-lingua.org/mecosim.