Adaptive Fuzzy Spiking Neural P Systems for Fuzzy Inference and Learning

Jun Wang[1] and Hong Peng[2]

[1] School of Electrical and Information Engineering,
Xihua University, Chengdu, Sichuan, 610039, China
[2] School of Mathematics and Computer Engineering,
Xihua University, Chengdu, Sichuan, 610039, China
wangjun@mail.xhu.edu.cn

Summary. Spiking neural P systems (in short, SN P systems) and their variants, including fuzzy spiking neural P systems (in short, FSN P systems), generally lack learning ability so far. Aiming at this problem, a class of modified FSN P systems are proposed in this paper, called adaptive fuzzy spiking neural P systems (in short, AFSN P systems). The AFSN P systems not only can model weighted fuzzy production rules in fuzzy knowledge base but also can perform dynamically fuzzy reasoning. It is more important that the AFSN P systems have learning ability like neural networks. Based on neuron's firing mechanisms, a fuzzy reasoning algorithm and a learning algorithm are developed. An example is included to illustrate the learning ability of the AFSN P systems.

Key words: Spiking neural P systems, Fuzzy spiking neural P systems, Adaptive fuzzy spiking neural P systems, Fuzzy reasoning, Learning pronlem

1 Introduction

Spiking neural P systems (in short, SN P systems) firstly introduced by Ionescu et al. in 2006 [1], are a class of distributed parallel computing models, which are incorporated into membrane computing from the way that biological neurons communicate through electrical impulses of identical form (spikes) [2]. Since then, a large number of SN P systems and their variants have been proposed [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. From the viewpoint of real-world applications, SN P systems have attractive due to the following features: (i) parallel computing advantage, (ii) high understandability (due to their directed graph structure), (iii) dynamic feature (neurons firing and spiking mechanisms make them suitable to model dynamic behaviors of a system), (iv) synchronization (that makes them suitable to describe concurrent events or activities), (v) non-linearly (that makes them suitable to process non-linear situation), and so on. Recently, in order to take full the advantage of SN P systems, a class of extended SN P systems were proposed by introducing fuzzy logic, which were called fuzzy spiking neural P systems (in short, FSN P systems) [12, 13, 14, 15]. The motivation of proposing these FSN P systems is to deal with the representation of fuzzy knowledge and model fuzzy reasoning in some real-world applications, such as process control, expert system, fault diagnosing, etc. As we know, since knowledge in real-world applications mentioned above is frequently updated, they are essentially dynamic systems. This requires that the FSN P systems should be adaptive, that at, FSN P systems must have ability to adjust themselves. However, the FSN P systems might fails to cope with potential changes of actual systems due to their lack of adaptive or learning mechanism. Besides, a few of adaptive SN P systems have been addressed in recent years [16, 17].

In this paper, we propose a class of modified FSN P systems, which are called adaptive fuzzy spiking neural P systems (in short, AFSN P systems). The practical motivation is to build a novel way to deal with the learning problem of dynamical fuzzy knowledge in some real-world applications under the framework of SN P systems. For this purpose, based on neuron's firing mechanisms, a fuzzy reasoning algorithm and a learning algorithm are developed in this paper.

The rest of this paper is organized as follows. In Section 2, we firstly present the AFSN P systems, and then describe a way to model weighted fuzzy production rules by the AFSN P systems, finally move on to give the developed fuzzy reasoning algorithm and learning algorithm. Simulation example is provided in Section 3. Finally, Section 4 draws the conclusions.

2 AFSN P Systems

2.1 Definition of AFSN P Systems

Currently, fuzzy spiking neural P systems (FSN P Systems, in short) have been discussed [12, 13, 14, 15]. However, they can not adjust themselves and lack learning ability. In this paper, we will introduce "adaptive" mechanism into the FSN P systems to propose a class of adaptive FSN P systems, called AFSN P systems.

Definition 1. *An AFSN P systems (*of degree $m \geq 1$*) is a construct of the form*

$$\Pi = (A, N_p, N_r, syn, I, O)$$

where

1) *A=$\{a\}$ is the singleton alphabet (the object a is called spike);*
2) *$N_p = \{\sigma_{p1}, \sigma_{p2}, \ldots, \sigma_{pm}\}$ is called proposition neuron set, where σ_{pi} is its i-th proposition neuron associated with a fuzzy proposition in weighted fuzzy production rules, $1 \leq i \leq m$. Each proposition neuron σ_{pi} has the form $\sigma_{pi} = (\alpha_i, \boldsymbol{\omega}_i, \lambda_i, r_i)$, where:*
   a) *$\alpha_i \in [0,1]$ and it is called the (potential) value of pulse contained in proposition neuron σ_{pi}. α_i is used to express fuzzy truth value of the proposition associated with proposition neuron σ_{pi}.*

   b) *$\boldsymbol{\omega}_i = (\omega_{i1}, \omega_{i2}, \ldots, \omega_{is_i})$ is called the output weight vector of the neuron σ_{pi}, where component $\omega_{ij} \in [0,1]$ is the weight on j-th output synapse (arc) of the neuron, $1 \le j \le s_i$, and s_i is the number of all output synapses (arc) of the neuron.*
   c) *r_i is a firing/spiking rule, of the form $E/a^\alpha \to a^\alpha$, where $\alpha \in [0,1]$. $E = \{\alpha \ge \lambda_i\}$ is called 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.*

3) *$N_r = \{\sigma_{r1}, \sigma_{r2}, \ldots, \sigma_{rn}\}$ is called rule neuron set, where σ_{ri} is its i-th rule neuron associated with a weighted fuzzy production rule, $1 \le i \le n$. Each rule neuron σ_{ri} has the form $\sigma_{ri} = (\alpha_i, \gamma_i, \tau_i, r_i)$, where*
   a) *$\alpha_i \in [0,1]$ is called the* (potential) value *of pulse contained in rule neuron σ_{ri}.*
   b) *$\gamma_i \in [0,1]$ is called the certain factor. It represents the strength of belief of the weighted fuzzy production rule associated with rule neuron σ_{ri}. At the same time, γ_i is also the weight on output synapse (arc) of the neuron.*
   c) *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 called 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.*

4) *$syn \subseteq (N_p \times N_r) \bigcup (N_r \times N_p)$ indicates synapses between both proposition neurons and rule neurons. Note that there are no synapse connections between any two proposition neurons or between any two rule neurons;*
5) *$I, O \subseteq N_p$ are input neuron set and output neuron set, respectively.*

In the AFSN P systems, there are two types of neurons: proposition neurons and rule neurons. In this paper, we denote proposition neurons and rule neurons by circles and rectangles respectively, shown in Fig. 1.

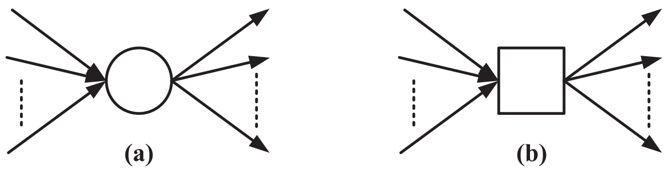

Fig. 1. Two types of neurons: (a) a proposition neuron; (b) a rule neuron.

For a proposition neuron, its content is used to express the fuzzy truth value of the fuzzy proposition associated with it. When its firing condition $E = \{\alpha \ge \lambda_i\}$ is satisfied, the neuron fires and its firing/spiking rule $E/a^\alpha \to a^\alpha$ can be applied. Applying the firing/spiking rule $E/a^\alpha \to a^\alpha$ means that the spike contained in the neuron is consumed, and then it produces a spike with value α, which will be weighted by the corresponding weight factor. Thus, its outputs are $\alpha \cdot \omega_i (i = 1, 2, \ldots, s)$.

Note that each rule neuron is assigned only an output weight ν. Suppose that a rule neuron has k predecessor proposition neurons. When it receives k spikes from its all predecessor proposition neurons and its firing condition $E = \{\alpha \geq \tau_i\}$ is satisfied, then it fires and its firing/spiking rule $E/a^{\alpha} \to a^{\beta}$ can be applied. The value of the received k spikes is calculated as its content α: $\alpha = x_1 + x_2 + \ldots + x_k$. Applying the firing/spiking rule $E/a^{\alpha} \to a^{\beta}$ means that the spike contained in the neuron is consumed, and then it produces a spike with value β where $\beta = \alpha \cdot \gamma$. Thus, its all outputs are $\alpha \cdot \gamma$.

Suppose that a proposition neuron has k predecessor rule neurons and it receives k spikes from them. Let output weights of the k predecessor rule neurons be $\gamma_1, \gamma_2, \ldots, \gamma_k$ respectively. If (potential) values of the received k spikes are $x_1, x_2, \ldots, x_k$ respectively, then its new content is computed by $\alpha = (x_1 + x_2 + \ldots + x_k)/(\gamma_1 + \gamma_2 + \ldots + \gamma_k)$.

2.2 Modeling Weighted Fuzzy Production Rules by AFSN P Systems

In many real-world applications such as expert system, fault diagnosing and process control, fuzzy production rules are used to describe the fuzzy relation between two propositions. In order to consider the degree of importance of each proposition in the antecedent contributing to the consequent, weighted fuzzy production rule has been introduced, and a more detained description can be found in [18, 19, 20].

However, we will discuss the following three types of weighted fuzzy production rules in order to study AFSN P systems in this paper.

Type 1: A simple fuzzy production rule

$$R: \text{ IF } p_1 \text{ THEN } p_2 \ (CF = \gamma), \ \tau, \ \omega$$

Type 2: A composite conjunctive rule

$$R: \text{ IF } p_1 \text{ AND } p_2 \text{ AND } \cdots \text{ AND } p_n \text{ THEN } p_{n+1} \ (CF = \gamma), \ \tau, \ \omega_1, \ \omega_2, \ldots, \omega_n$$

Type 3: A composite disjunctive rule

$$R: \text{ IF } p_1 \text{ OR } p_2 \text{ OR } \cdots \text{ OR } p_n \text{ THEN } p_{n+1} \ (CF = \gamma), \ \tau, \ \omega_1, \ \omega_2, \ldots, \omega_n$$

Above three types of weighted fuzzy production rules can be modeled by the proposed AFSN P systems according to the idea that each fuzzy proposition is mapped into one proposition neuron and each fuzzy production rule is mapped into one rule neuron or several rule neurons. Thus, the three types of weighted fuzzy production rules are represented by the following three AFSN P systems, Π_1, Π_2 and Π_3, respectively:

- $\Pi_1 = (A, \{\sigma_{p1}, \sigma_{p2}\}, \{\sigma_{r1}\}, syn, I, O)$ where:
  (1) $A = \{a\}$
  (2) For each j $(j = 1, 2)$, $\sigma_{pj} = (\alpha_j, \boldsymbol{\omega}, \lambda, r_j)$ is a proposition neuron associated with proposition p_j, and r_j is a spiking rule of the form $E/a^{\alpha} \to a^{\alpha}$.

(3) $\sigma_{r1} = (\alpha_3, \gamma, \tau, r_3)$ is a rule neuron associated with rule R, and r_3 is a spiking rule of the form $E/a^{\alpha} \rightarrow a^{\beta}$.
(4) $syn = \{(\sigma_{p1}, \sigma_{r1}), (\sigma_{r1}, \sigma_{p2})\}$.
(5) $I = \{\sigma_{p1}\}$, $O = \{\sigma_{p2}\}$.

Fig. 2(a) shows the AFSN P system model of *Type 1*: Π_1.

- $\Pi_2 = (A, \{\sigma_{p1}, \sigma_{p2}, \ldots, \sigma_{pn}, \sigma_{p(n+1)}\}, \{\sigma_{r1}\}, syn, I, O)$ where:
(1) $A = \{a\}$
(2) For each j ($j = 1, \ldots, n, n+1$), $\sigma_{pj} = (\alpha_j, \boldsymbol{\omega}_j, \lambda_j, r_j)$ is a proposition neuron associated with proposition p_j, and r_j is a spiking rule of the form $E/a^{\alpha} \rightarrow a^{\alpha}$.
(3) $\sigma_{r1} = (\alpha_{n+2}, \gamma, \tau, r_{n+2})$ is a rule neuron associated with rule R, and r_{n+2} is a spiking rule of the form $E/a^{\alpha} \rightarrow a^{\beta}$.
(4) $syn = \{(\sigma_{p1}, \sigma_{r1}), (\sigma_{p2}, \sigma_{r1}), \ldots, (\sigma_{pn}, \sigma_{r1}), (\sigma_{r1}, \sigma_{p(n+1)})\}$.
(5) $I = \{\sigma_{p1}, \sigma_{p2}, \ldots, \sigma_{pn}\}$, $O = \{\sigma_{p(n+1)}\}$.

Fig. 2(b) shows the AFSN P system model of *Type 2*: Π_2 (in the case of $n = 2$).

- $\Pi_3 = (A, \{\sigma_{p1}, \sigma_{p2}, \ldots, \sigma_{pn}, \sigma_{p(n+1)}\}, \{\sigma_{r1}, \sigma_{r2}, \ldots, \sigma_{rn}\}, syn, I, O)$ where:
(1) $A = \{a\}$
(2) For each j ($j = 1, \ldots, n, n+1$), $\sigma_j = (\alpha_j, \boldsymbol{\omega}_j, \lambda_j, r_j)$ is a proposition neuron associated with proposition p_j, and r_j is a spiking rule of the form $E/a^{\alpha} \rightarrow a^{\alpha}$.
(3) For each j ($j = 1, \ldots, n$), $\sigma_{rj} = (\alpha_{n+j+1}, \gamma_j, \tau_j, r_{n+j+1})$ is a rule neuron associated with rule R, and r_{n+j+1} is a spiking rule of the form $E/a^{\alpha} \rightarrow a^{\beta}$.
(4) $syn = \{(\sigma_{p1}, \sigma_{r1}), (\sigma_{p2}, \sigma_{r2}), \ldots, (\sigma_{pn}, \sigma_{rn}), (\sigma_{r1}, \sigma_{p(n+1)}), (\sigma_{r2}, \sigma_{p(n+1)}), \ldots, (\sigma_{rn}, \sigma_{p(n+1)})\}$.
(5) $I = \{\sigma_{p1}, \sigma_{p2}, \ldots, \sigma_{pn}\}$, $O = \{\sigma_{p(n+1)}\}$.

Fig. 2(c) shows the AFSN P system model of *Type 3*: Π_3 (in the case of $n = 2$).

2.3 Fuzzy Reasoning Based on AFSN P systems

Since the presented AFSN P systems mainly focus on the weighted fuzzy reasoning, we assume that firing threshold of every proposition neuron is $\lambda = 0$. This means that once a proposition neuron contains a spike with $\alpha > 0$ it will fires. According to firing mechanism of AFSN P systems, fuzzy reasoning processes of above three types of weighted fuzzy production rules can be described as follows:

- For *Type 1*, we can set $\omega = 1$ since there is only one proposition in antecedent of the rule R. Initially, assume that neuron σ_{p1} contains a spike with $\alpha_1 > 0$. At first step, neuron σ_{p1} fires and emits a spike with α_1. At second step, neuron σ_{r1} receives the spike. If $\alpha_1 \geq \tau$, then neuron σ_{r1} fires and emits a spike with

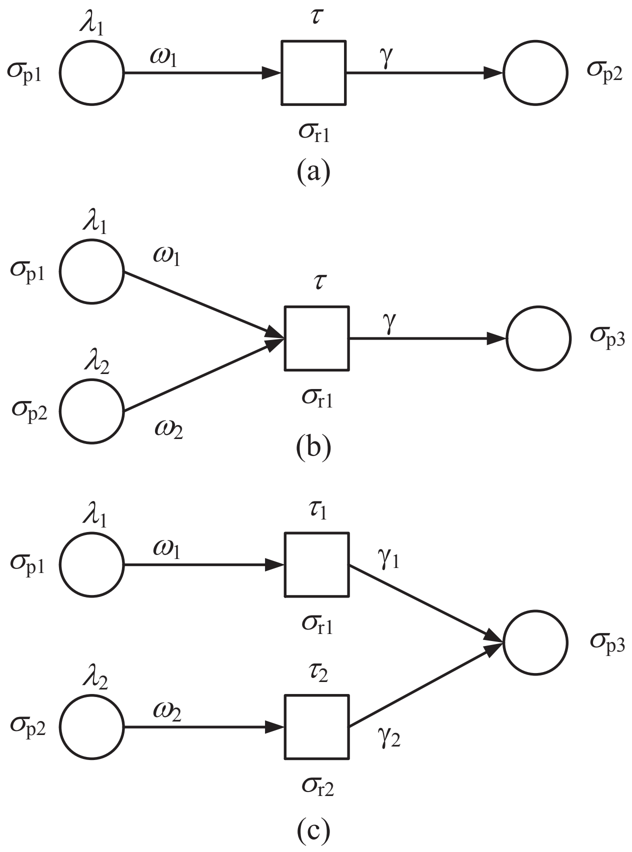

Fig. 2. AFSN P systems of weighted fuzzy production rules of three types: (a) *Type 1*; (b) *Type 2*; (c) *Type 3*.

$\alpha_1 \cdot \gamma$. Neuron σ_{p2} will receive the spike at next step. Thus, α_2 can be expressed by

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

- For *Type 2*, assume that neurons $\sigma_{p1}, \sigma_{p2}, \ldots, \sigma_{pn}$ contain a spike with $\alpha_1 > 0, \alpha_2 > 0, \ldots, \alpha_n > 0$, respectively. At first step, the n neuron fire simultaneously, and emit a spike with $\alpha_1, \alpha_2, \ldots, \alpha_n$, respectively. At second step, neuron σ_{r1} receives the n spikes and its content is updated as $\sum_{i=1}^{n} \alpha_i \cdot \omega_i$. If $(\sum_{i=1}^{n} \alpha_i \cdot \omega_i) \geq \tau$, then neuron σ_{r1} fires and emits a spike with $(\sum_{i=1}^{n} \alpha_i \cdot \omega_i) \cdot \gamma$. Neuron $\sigma_{p(n+1)}$ will receive the spike at next step. Thus, α_{n+1} can be expressed by

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

- For *Type 3*, we can set $\omega_1 = \omega_2 = 1$. Assume that neurons $\sigma_{p1}, \sigma_{p2}, \ldots, \sigma_{pn}$ contain a spike with $\alpha_1 > 0, \alpha_2 > 0, \ldots, \alpha_n > 0$, respectively. At first step, the n neuron fire simultaneously, and emit a spike with $\alpha_1, \alpha_2, \ldots, \alpha_n$, respectively. At second step, each neuron σ_{ri} receives a spike sent by σ_{pi}, whose value is α_i, $i = 1, 2, \ldots, n$. Let $J = \{j \mid \alpha_j \geq \tau_j, j = 1, 2, \ldots, n\}$. Then neurons $\sigma_{rj}(j \in J)$ fire and each neuron of them emits a spike. Neuron $\sigma_{p(n+1)}$ will receive the spikes at next step. Thus, α_{n+1} can be expressed by

$$\alpha_{n+1} = \begin{cases} \Big(\sum\limits_{j \in J} \alpha_j \cdot \gamma_j\Big) \Big/ \Big(\sum\limits_{j \in J} \gamma_j\Big), & \text{if } \alpha_j \geq \tau_j, j \in J \\ 0, & \text{if } \alpha_j < \tau_j, j = 1, 2 \ldots, n \end{cases} \tag{3}$$

From fuzzy reasoning process described above, we can see that fuzzy reasoning based on AFSN P systems are easily implemented. Thus, through firing mechanism of AFSN P systems, certainty factors can be reasoned from a set of known antecedent propositions to a set of consequent propositions step by step.

Let $P_{current} = \{\sigma_{pi} \mid \sigma_{pi} \in N_p, \alpha_i > 0\}$ be a set of current enabled proposition neurons. If a neuron $\sigma_{pi} \in P_{current}$, then it will fire. Let $R_{current} = \{\sigma_{rj} \mid \sigma_{rj} \in N_r, \alpha_j > \tau_j\}$ be a set of current enabled rule neurons. Likewise, if a neuron $\sigma_{rj} \in R_{current}$, then it will fire. Therefore, fuzzy reasoning algorithm based on AFSN P systems can be summarized as follows.

```
program Fuzzy_reasoning_algorithm
input
   Certainty factors of a set of antecedent propositions, which
      are corresponding to I of AFSN P systems;
output
   Certainty factors of a set of consequence propositions, which
      are corresponding to O of AFSN P systems;
begin
   Pcurrent := I;
   Rcurrent := {}
   P := Np;
   R := Nr;
   repeat
      Compute the outputs of current enabled proposition
         neurons in Pcurrent;
      Find current enabled rule neurons Rcurrent form R;
      Compute the outputs of current enabled proposition
         neurons in Rcurrent;
      P := P - Pcurrent;
```

```
        R := R - Rcurrent;
        Find current enabled proposition neurons Pcurrent form P;
    until P = {} and R = {}
end.
```

2.4 Learning of AFSN P systems

In order to deal with the learning problem of AFSN P systems, we assume that

1) AFSN P system model Π has been developed;
2) In the AFSN P system model, weights and thresholds of all rule neurons are known;
3) Certainty factor values of all neurons in I and O are given.

From the discussion above, we know that the presented AFSN P systems are mainly used to model weighted fuzzy production rules and these rules consist of three types. So, an AFSN P system model can be divided into three types of sub-structures, which are shown in Fig.2(a)-(c). Therefore, the learning of entire system can be decomposed to several simpler learning procedures of the sub-nets. This means that the complexity of the learning algorithm can be greatly reduced. According to above assumption, certainty factors of the proposition neurons associated with antecedent propositions are known, however, their weights are unknown. Therefore, these weights need to be learned. Note that for AFSN P system Π_1 of *Type 1*, we have $\omega_1 = 1$, while we have $\omega_1 = \omega_2 = 1$ for AFSN P system Π_3 of *Type 3*. So, only weights of AFSN P system Π_2 of *Type 2* need to be learned. In order to carry out the weight learning, the AFSN P system Π_2 of *Type 2* can be converted to a single-layer neural network, shown in Fig.3. So, Widrow-Hoff learning law (Least Mean Square) can be applied in this paper.

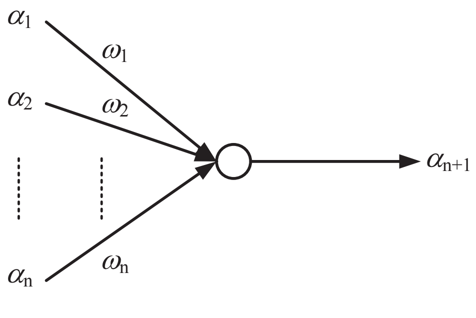

Fig. 3. The single-layer neural network converted by the AFSN P system Π_2 of *Type 2*.

We can summarize the learning algorithm of AFSN P systems as follows

```
program Weight_learning_algorithm
input
   Training data set D;
   m = |D|;
   Learning rate delta;
output
   The weights (w1, w2,..., wn);
begin
   Select a set of initial weights;
   i=1;
   repeat
      Compute the outputs error of i-th training sample;
      Update the weights (w1, w2,..., wn) using Widrow-Hoff
         learning law with learning rate delta;
      i = i + 1
   until i>m
end.
```

3 Simulation

In this section, a typical example is selected to illustrate the learning ability.

Example 1. Let p_1, p_2, p_3, p_4, p_5 and p_6 are related propositions of a knowledge base of fault diagnosis. There are the following weighted fuzzy production rules:
R_1: IF p_1 THEN p_4 (γ_1, τ_1)
R_2: IF p_2 AND p_4 THEN p_5 (ω_2, ω_4, γ_2, τ_2)
R_3: IF p_3 AND p_5 THEN p_6 (γ_3, γ_4, τ_3, τ_4)

This example includes three types of rules: R_1 is a simple rule and R_2 is a composite conjunctive rule, while R_3 is a composite disjunctive rule. These weighted fuzzy production rules can be modeled by the following AFSN P system Π:

- $\Pi = (A, \{\sigma_{p1}, \sigma_{p2}, \sigma_{p3}, \sigma_{p4}, \sigma_{p5}, \sigma_{p6}\}, \{\sigma_{r1}, \sigma_{r2}, \sigma_{r3}, \sigma_{r4}\}, syn, I, O)$
  where:
  (1) $A = \{a\}$
  (2) For each j ($j = 1, 2, 3, 4, 5, 6$), $\sigma_{pj} = (\alpha_j, \boldsymbol{\omega}_j, \lambda_j, r_j)$ is a proposition neuron associated with proposition p_j, and r_j is a spiking rule of the form $E/a^\alpha \rightarrow a^\alpha$. Here, $\lambda_j(j = 1, 2, \ldots, 6) = 0$, and $\omega_1 = \omega_3 = \omega_5 = 1$.
  (3) For each j ($j = 1, 2, 3, 4$), $\sigma_{rj} = (\alpha_{k+j}, \gamma_j, \tau_j, r_{k+j})$ is rule neuron. σ_{r1} and σ_{r2} are associated with rule R_1 and R_2 respectively, while σ_{r3} and σ_{r4} are associated with rule R_3. $r_{k+j}(j = 1, 2, 3, 4)$ are spiking rule of the form $E/a^\alpha \rightarrow a^\beta$.
  (4) $syn = \{(\sigma_{p1}, \sigma_{r1}), (\sigma_{p2}, \sigma_{r2}), (\sigma_{p3}, \sigma_{r3}), (\sigma_{p4}, \sigma_{r2}), (\sigma_{p5}, \sigma_{r4}), (\sigma_{r1}, \sigma_{p4}), (\sigma_{r2}, \sigma_{p5}), (\sigma_{r3}, \sigma_{p6}), (\sigma_{r4}, \sigma_{p6})\}$.
  (5) $I = \{\sigma_{p1}, \sigma_{p2}, \sigma_{p3}\}$, $O = \{\sigma_{p4}, \sigma_{p5}, \sigma_{p6}\}$.

Fig.4 shows the AFSN P system Π. The AFSN P system has three input proposition neurons $\{\sigma_{p1}, \sigma_{p2}, \sigma_{p3}\}$ and three output proposition neurons $\{\sigma_{p4}, \sigma_{p5}, \sigma_{p6}\}$. Suppose parameters of the AFSN P system are given as follows:

$$\begin{array}{c} \gamma_1 = 0.80,\ \gamma_2 = 0.85,\ \gamma_3 = 0.85,\ \gamma_4 = 0.90 \\ \tau_1 = 0.40,\ \tau_2 = 0.60,\ \tau_3 = 0.55,\ \tau_4 = 0.45 \end{array} \tag{4}$$

Here, weights ω_2 and ω_4 are unknown. Assume the ideal weights are $\omega_2^* = 0.63$ and $\omega_4^* = 0.37$. Using fuzzy reasoning algorithm, we can obtain a set of output data (certainty factors of consequence propositions) according to the input data (certainty factors of antecedent propositions). Table 1 gives the part of the reasoning results of the AFSN P system.

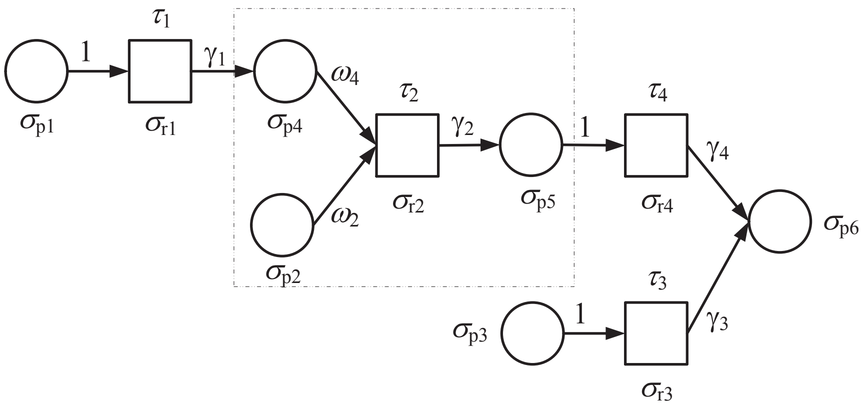

Fig. 4. AFSN P system of *Example* 1.

Table 1. The reasoning results of AFSN P systems.

No.	α_1	α_2	α_3	α_4	α_5	α_6
1	0.8762	0.7724	0.8536	0.7010	0.6341	0.7470
2	0.8325	0.8271	0.6124	0.6660	0.6524	0.6365
3	0.7518	0.8912	0.5896	0.6390	0.6782	0.6326
4	0.6785	0.7216	0.6518	0.5767	0.5678	0.6110
5	0.6127	0.6874	0.7829	0.5208	0.5319	0.6610
6	0.5866	0.8516	0.5908	0.4986	0.6128	0.6015
7	0.5236	0.7835	0.5862	0.4451	0.5595	0.5732
8	0.3645	0.7845	0.6628	0.0	0.0	0.3409
9	0.5235	0.5648	0.7461	0.4450	0.0	0.3837
10	0.3246	0.6324	0.5582	0.0	0.0	0.2871
...	...	...	...	...	...	...

From Fig.4, we can see that only two weights ω_2 and ω_4 need to be learned in the AFSN P system Π. In this paper, neural network technique will be employed to adjust the two weights. The learning part of the AFSN P system Π (see the part in the dashed box of Fig.4) can be transformed as a single layer neural network (see Fig.5):

$y(t) = W(t)^T X(t) + b$

where t is time, $X(t) = [\alpha_2(t), \alpha_4(t)]^T$ is input vector, $W(t) = [\omega_2(t), \omega_4(t)]^T$ is weights vector, and b is the bias.

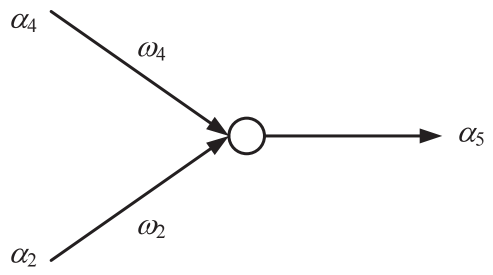

Fig. 5. The neural network transformed by the learning part in the AFSN P system of *Example* 1.

In order to learn these weights by using neural networks, Widrow-Hoff learning law can be applied as follows

$$W(t+1) = W(t) + 2\delta e(t) X(t), \tag{5}$$

$$e(t) = y^*(t) - y(t) \tag{6}$$

Here, we select $\delta = 1.23$. Let initial weights be $W(0) = [\omega_2(0), \omega_4(0)]^T = [0.5, 0.2]^T$. By applying Widrow-Hoff learning law, after a training process ($t > 33$), the two weights convergence to their real values. Fig.6 shows simulation results.

Form the example, we can see that the fuzzy reasoning algorithm and the Widrow-Hoff learning are very effective if we do not know the weights of AFSN P systems. After a training process, we can build a good input-output mapping relation of a knowledge system.

4 Conclusion

In this paper, we presented a class of modified fuzzy spiking neural P systems: adaptive fuzzy spiking neural P systems (AFSN P systems, in short). In addition to fuzzy knowledge representation and dynamically fuzzy reasoning, they have learning ability as neural netwarks. Therefore, fuzzy knowledge in knowledge base not only can be modeled by a AFSN P system but also can be learning through the

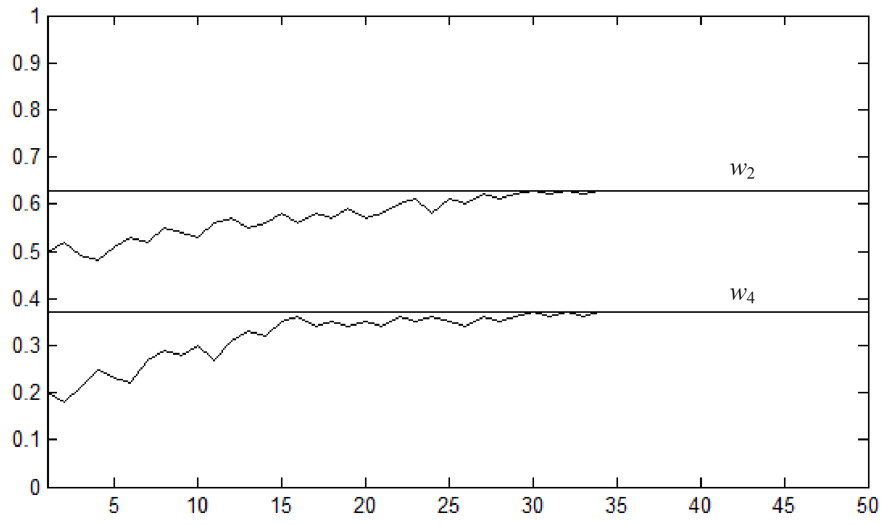

Fig. 6. The weight learning results of *Example* 1.

AFSN P system. The results presented in this paper provide a novel way to solve the knowledge learning problem in some real-world applications, such as expert systems, fault diagnosis, process control, and so on.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (Grant No. 61170030), Research Fund of Sichuan Provincial Key Discipline of Power Electronics and Electric Drive, Xihua University (No. SZD0503-09-0), Research Fund of Sichuan Key Laboratory of Intelligent Network Information Processing (No. SGXZD1002-10), and the Importance Project Foundation of Xihua University (No. Z1122632), China.

References

1. Ionescu, M., Păun, Gh., Yokomori, T.: Spiking Neural P Systems. Fundameta Informaticae 71(2-3), 279–308 (2006)
2. Păun, Gh., Rozenberg, G., Salomaa, A.: The Oxford Handbook of Membrance Computing. Oxford Unversity Press, New York (2010)
3. Păun, Gh., Pérez-Jiménez, M.J., Rozenberg, G.: Spike Train in Spiking Neural P Systems. Int. J. Found. Comp. Sci. 17(4), 975–1002 (2006)

4. Chen, H., Ishdorj, T.-O., Păun, Gh., Perez-Jimenez, M.J.: Handling Languages with Spiking Neural P Systems with Extended Rules. Romanian Journal of Information Science and Technology 9(3), 151–162 (2006)
5. Chen, H., Ishdorj, T.-O., and Gh. Păun, Computing Along The Axon. Progress in Natural Science 17(4), 417–423 (2007)
6. Ionescu, M., Păun, Gh., Yokomori, T.: Spiking Neural P Systems with An Exhaustive Use of Rules. Int. J. of Unconvent. Comt. 3(2), 135–154 (2007)
7. Freund, R., Ionescu, M., Oswald, M.: Extended Spiking Neural P Systems with Decaying Spikes and/or Total Spiking. Int. J. of Foundations of Computer Science 19(5), 1223–1234 (2008)
8. Pan, L., Păun, G.: Spiking Neural P Systems with Anti-Spikes. Int. J. of Computers, Communications & Control, 4(3), 273-282 (2009)
9. Wang, J., Hoogeboom, H.J., Pan, L., Păun, Gh.: Spiking Neural P Systems with Weights and Thresholds. In: Proceedings of Tenth Workshop on Membrane Computing (WMC10), August 2009, pp. 514-533 (2009)
10. Cavaliere, M. Ibarra, O.H., Păun, G., Egecioglu, O., Ionescu, M., Woodworth, S. Asynchronous Spiking Neural P Systems. Theoretical Computer Science 410(24-25) 2352–2364 (2009)
11. Pan, L., Păun, G.: Spiking Neural P Systems: An Improved Normal Form. Theoretical Computer Science 411(6) 906–918 (2010)
12. Wang, J., Peng, H.: Fuzzy Knowledge Representation Based on An Improving Spiking Neural P System. In: 2010 Sixth International Conference of Natural Computing, ICNC2010, 6, 3012–3015 (2010)
13. Wang, T., Wang, J. Peng, H. Deng, Y.L.: Knowledge Representation Using Fuzzy Spiking Neural P System. In: 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications, Volume-1, 586–590 (2010)
14. Wang, J., Zhou, L., Peng, H., Zhang, G.X.: An Extended Spiking Neural P System for Fuzzy Knowledge Representation. International Journal of Innovative Computing, Information and Control, 7(7A), 3709–3724 (2011)
15. Peng, H., Wang, J., Perez-Jimenez, M.J., Wang, H., Shao, J., Wang, T.: Fuzzy Reasoning Spiking Neural P System for Fault Diagnosis. Information Sciences, 2012 (Accepted)
16. Gutierrez-Naranjo, M.A., Perez-Jimenez, M.J.: Hebbian Learning from Spiking Neural P Systems View. Lecture Notes in Computer Science, Volume 5391/2009, 217–230 (2009)
17. Peng, H., Wang, J.: Adaptive Spiking Neural P Systems. In: 2010 Sixth International Conference of Natural Computing, ICNC2010, 6, 3008–3011 (2010)
18. Yeung, D.S., Tsang, E.C.C.: Weighted Fuzzy Production Rules. Fuzzy Sets and Systems, 88, 299–313 (1997)
19. Yeung, D.S., Tsang, E.C.C.: A Multilevel Weighted Fuzzy Reasoning Algorithm for Expert Systems. IEEE Trans. Syst., Man, Cybern. A, 28(2), 149–158 (1998)
20. Chen, S.-M., Ko, Y.-K., Chang, Y.-C., Pan J.-S.: Weighted Fuzzy Interpolative Reasoning Based on Weighted Increment Transformations and Weighted Ratio Transformation Techniques. IEEE Transactions on Fuzzy Systems, 17(6), 1412–1427 (2009)