Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Sciences 170 (2005) 419–441
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

Reasoning and unsupervised learning in a fuzzy cognitive map

Amit Konar [a], Uday K. Chakraborty [b,*]

[a] *Electronics and Tele-Communication Engineering Department, Jadavpur University, Calcutta 700032, India*

[b] *Math & Computer Science Department, University of Missouri, One University Boulevard, St. Louis, MO 63121, USA*

Received 2 July 2003; received in revised form 8 March 2004; accepted 9 March 2004

Abstract

This paper presents a new model for unsupervised learning and reasoning on a special type of cognitive maps realized with Petri nets. The unsupervised learning process in the present context adapts the weights of the directed arcs from transitions to places in the Petri net. A Hebbian-type learning algorithm with a natural decay in weights is employed to study the dynamic behavior of the algorithm. The algorithm is conditionally stable for a suitable range of the mortality rate. After convergence of the learning algorithm, the network may be used for computing the beliefs of the desired propositions from the supplied beliefs of the axioms (places with no input arcs). Because of the conditional stability of the algorithm, it may be used in complex decision-making and learning such as automated car driving in an accident-prone environment. The paper also presents a new model for knowledge refinement by adaptation of weights in a fuzzy Petri net using a different form of Hebbian learning. This second model converges to stable points in both encoding and recall phases.

1. Introduction

Kosko's [15,16] pioneering work on fuzzy cognitive maps stands as a milestone in the field. However, his model has two limitations. First, the model

* Corresponding author. Tel.: +1-314-516-6339; fax: +1-314-516 5400.
E-mail address: uday@cs.umsl.edu (U.K. Chakraborty).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2004.03.012

cannot describe many-to-one (or -many) causal relations. Second, the recall model gets trapped within limit cycles and is therefore not applicable in real-time decision-making problems. In this paper we propose an extension of Kosko's model that can represent many-to-one (or -many) causal relations by using Petri nets. The proposed recall model is free from limit cyclic behavior.

In the proposed model, many-to-one (or -many) causal relations are described as a joint occurrence of a set of antecedent clauses and one or more disjunctive consequent clauses. Further, instead of representing positive and negative causal relations separately, we can describe them by a uniform notion without attaching any "sign" label to the arcs in the fuzzy cognitive map. To see this, suppose there is a negative causal relation:

$$P \xrightarrow{-} Q$$

between concepts P and Q, which means P causally decreases Q. We can represent the same causal relation as

$$P \rightarrow \neg Q$$

which means P causally increases the dis-concept Q. The $\neg$ symbol is a logical negation operator. Thus we can represent all negative causal relations as equivalent positive causal relations without having to attach signs to the arcs to label the positive/negative causality.

Another point that needs to be addressed before formally introducing the scheme is: Why do we select fuzzy Petri nets (FPNs) [1–14,17,22–26] to model fuzzy cognitive maps? The answer to this is that FPNs support the necessary many-to-one (or -many) causal relations and have already proved themselves useful as an important reasoning [9–14] and learning [22] tool. The principle of reasoning and learning that we introduce in this paper, however, is new and is different from the existing works.

The paper is organized as follows. The encoding and recall model of the proposed cognitive map is introduced in Section 2. The state-space representation of the model is given in Section 3. Section 4 presents an analysis of stability of the proposed model. A computer simulation of an automated collision-free car driving system based on the proposed model is presented in Section 5. Section 6 summarizes the significance of the results. Section 7 presents another model—a new technique for estimation of certainty factors, including convergence proofs of both encoding and recall phases and an illustrative application. Conclusions are drawn in Section 8.

2. The proposed model for cognitive reasoning

The proposed model for cognitive reasoning has two main components: (i) the encoding and (ii) the recall modules. The encoding is needed to store stable fuzzy weights at the arcs, connected between transitions and places [17], of an FPN, while the recall is required for the purpose of deriving stable fuzzy inferences. Before presenting the model, we formally define an FPN [18].

Definition 1. An FPN is a 9-tuple, given by

$$\mathrm{FPN} = \langle P, \mathrm{Tr}, T, D, I, O, \mathrm{Th}, n, W \rangle \tag{1}$$

where

$P = (p_1, p_2, \ldots, p_n)$ is a finite set of places;
$\mathrm{Tr} = (\mathrm{tr}_1, \mathrm{tr}_2, \ldots, \mathrm{tr}_m)$ is a finite set of transitions;
$T = (t_1, t_2, \ldots, t_m)$ is a set of tokens in the interval $[0, 1]$ associated with the transitions $(\mathrm{tr}_1, \mathrm{tr}_2, \ldots, \mathrm{tr}_m)$ respectively;
$D = (d_1, d_2, \ldots, d_n)$ is a finite set of propositions, proposition d_k corresponds to place p_k;
$P \cap \mathrm{Tr} \cap D = \emptyset$; cardinality of (P) = cardinality of (D);
$I : \mathrm{Tr} \to P^{\propto}$ is the input function, representing a mapping from transitions to bags of (their input) places;
$O : \mathrm{Tr} \to P^{\propto}$ is the output function, representing a mapping from transitions to bags of (their output) places;
$\mathrm{Th} = (\mathrm{th}_1, \mathrm{th}_2, \ldots, \mathrm{th}_m)$ represents a set of threshold values in the interval $[0, 1]$ associated with transitions $(\mathrm{tr}_1, \mathrm{tr}_2, \ldots, \mathrm{tr}_m)$ respectively;
$n : P \to [0, 1]$ is an association function hereafter called fuzzy belief, representing a mapping from places to real values between 0 and 1; $n(p_i) = n_i$ (say); and
$W = \{w_{ij}\}$ is the set of weights from the jth transition to the ith place, where i and j are integers.

We now present an example to illustrate the above parameters in a typical FPN structure.

Example 1. The FPN in Fig. 1 represents the following causal rule (CR):

$$\mathrm{CR} : (it - is - hot), (the - sky - is - cloudy) \to (it - will - rain).$$

The above rule means that the events (concepts) "it-is-hot" and "the-sky-is-cloudy" causally influence the event "it-will-rain". For all practical purposes, this rule is identical to a typical If–Then relation.

Here $P = \{p_1, p_2, p_3\}$, $\mathrm{Tr} = \{\mathrm{tr}_1\}$, $T = \{t_1\}$, $D = \{d_1, d_2, d_3\}$ where d_1 = it-is-hot, d_2 = the-sky-is-cloudy and d_3 = it-will-rain. $I(\mathrm{tr}_1) = \{p_1, p_2\}$, $O(\mathrm{tr}_1) = \{p_3\}$, $\mathrm{Th} = \{\mathrm{th}_1\}$, $n = \{n_1, n_2, n_3\}$ and $w = \{w_{31}\}$.

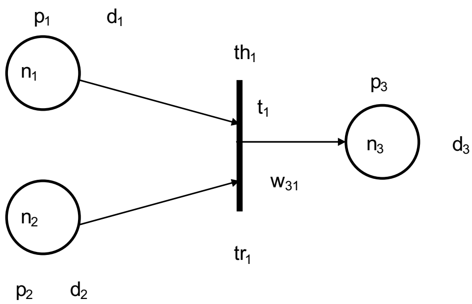


Fig. 1. A fuzzy Petri net representing the given causal rule.

2.1. Encoding of weights

The encoding model, represented by Eq. (2), has been designed based on the Hebbian learning concept [27], which states that *the strength of the weight of an arc should increase with increasing signal levels at the adjacent nodes (here places and transitions)*.

$$\frac{\mathrm{d}w_{ij}(t)}{\mathrm{d}t} = -\alpha w_{ij}(t) + t_j(t)n_i(t) \tag{2}$$

where w_{ij} denotes the weight of the arc connected between transition tr_j and place p_i, α represents the forgetfulness (mortality) rate, t_j represents the token associated with transition tr_j (also called truth token) and n_i represents the fuzzy belief associated with proposition d_i. The $-\alpha w_{ij}(t)$ term indicates a natural decay of the weight w_{ij} in the cognitive map. This is similar to the natural tendency of forgetfulness of human beings.

The discrete version of expression (2) takes the following form:

$$w_{ij}(t+1) = (1-\alpha)w_{ij}(t) + t_j(t)\cdot n_i(t) \tag{3}$$

2.2. The recall model

The recall process aims at updating fuzzy beliefs at the places. It is important to note that the recall model (to be proposed shortly) appears in both encoding and recall phases of cognitive learning. A question that naturally arises: why does the recall model become part of an encoding phase? The answer to this is presented below. The adaptation of weights by the recursive equation (3) strengthens some weights and weakens some weights. This means that some concepts should be strengthened and some weakened. A similar *metamorphosis* of concepts always takes place in the mind of human beings. This justifies the inclusion of the recall model in the encoding phase. On the other hand, once the weights are stabilized, the recall/reasoning model is needed in its own course of action to derive new fuzzy inferences from the known beliefs of a set of axioms (places with no input arcs).

Before describing the recall of fuzzy beliefs at places, we define the firing condition of a transition. *A transition fires if the ANDed value of fuzzy beliefs of its input places exceeds the threshold associated with it*. On firing of a transition, the ANDed (min) value of beliefs of its input places is transmitted to its output [18]. This is described below by expression (4) (vide Fig. 2):

$$t_q(t+1) = \left(\bigwedge_{k=1}^{n} n_k(t) \right) \wedge u\left[\left(\bigwedge_{\exists k=1}^{n} n_k(t) \right) - \mathrm{th}_q \right] \tag{4}$$

where $n_k(t)$ represents the fuzzy belief of place p_k (an input place of transition tr_q) and u is a unit step function. "$\wedge$" denotes fuzzy AND (min) operator. In the present context, the first $\wedge$ operator takes the minimum of the beliefs n_k of the corresponding input places p_k in the FPN. The u operator checks the firing condition of the transition. If the transition fires, u returns a one (1) value, else it is zero (0). The second $\wedge$ operator combines the first part of token computation with the third part of firing condition checking. The third $\wedge$ operator is self-explanatory. The notation $\exists k$ below the third $\wedge$ operator represents that the transition tr_q possesses at most n number of input places.

Belief updating at place p_i is examined below by expression (5) (vide Fig. 3):

$$n_i(t+1) = n_i(t) \vee \left(\bigvee_{j=1}^{m} (w_{ij}(t) \wedge t_j(t+1)) \right) \tag{5}$$

where $w_{ij}(t)$ denotes the weight of the arc connected between transition tr_j and place p_i. Here, $n_i(t+1)$ is computed by taking the maximum ($\vee$) of the current belief $n_i(t)$ and the influence from the other transitions that can directly affect the belief of place p_i.

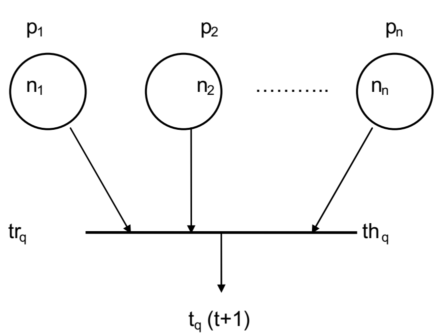


Fig. 2. An FPN representing connectivity from n places to transition tr_q.

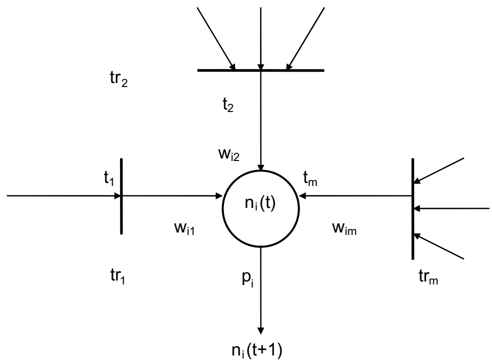


Fig. 3. An FPN representing connectivity from m transitions to a place p_i.

It may be added here that during the recall cycle, fuzzy truth tokens (FTT) at all transitions are updated concurrently, and the results, thus obtained, are used for computing beliefs at all places in parallel by using expression (5). This is referred to as the *belief revision/recall cycle*.

3. State-space formulation

The state-space formulation is required for representing the scheme of cognitive reasoning by using simple arrays instead of graphs for implementation. We will develop three expressions for updating FTT of transitions, fuzzy belief at places and weights in vector–matrix form with the help of two binary connectivity matrices P and Q, defined below.

Definition 2. A *place to transition connectivity* (*PTC*) *matrix* Q is a binary matrix whose element $q_{jk} = 1$, if for place p_k and transition tr_j, $p_k \in I(\text{tr}_j)$; otherwise $q_{jk} = 0$. If the FPN has n places and m transitions, then the Q matrix is of dimension $(m \times n)$ [10].

Definition 3. A *transition to place connectivity* (*TPC*) *matrix* P is a binary matrix whose element $p_{ij} = 1$, if for place p_i and transition tr_j, $p_i \in O(\text{tr}_j)$; otherwise $p_{ij} = 0$, With n places and m transitions in the FPN, the P matrix is of dimension $(n \times m)$ [11].

3.1. State-space model for belief updating

The equation for belief updating at places is already presented in expression (5). For n places in the FPN, there will be n expressions analogous to (5).

Further, if we have m transitions in the FPN, not all of them may be connected to place p_i. So, to represent connections from at least one of m number of transitions to a place p_i, we use connectivity matrix W and thus j is replaced by $\exists j$ (some j) in the range 1 to m. Thus we find the following expression:

$$n_i(t+1) = n_i(t) \vee \left[\bigvee_{\exists j=1}^{m} (t_j(t+1) \wedge w_{ij}) \right] \qquad \text{(revised (5))}$$

For all n places, the expression presented above takes the following form:

$$N(t+1) = N(t) \vee (W \circ T(t+1)) \qquad (6)$$

where N = fuzzy belief vector of dimension $(n \times 1)$, the rth component of which represents fuzzy belief of place p_r, W = fuzzy weight matrix from places to their input transitions (dimension $(m \times n)$) and T = token vector for transitions (dimension $(m \times 1)$), such that the rth component of T represents the fuzzy truth token of the rth transition. The operator "$\circ$" represents fuzzy max–min composition operator, which is executed analogous to conventional matrix multiplication algebra, with the replacement of algebraic multiplication by fuzzy AND (min) and addition by fuzzy OR (max) operator.

3.2. State-space model for FTT updating of transitions

Applying De Morgan's law, the transition updating equation described by expression (4) can be re-modeled as follows:

$$t_q(t+1) = \left(\bigvee_{\exists k=1}^{n} (n_k^{\mathrm{c}}) \right)^{\mathrm{c}} \wedge u\left[\left(\bigvee_{\exists k=1}^{n} (n_k^{\mathrm{c}}) \right)^{\mathrm{c}} - \mathrm{th}_q \right] \qquad \text{(revised (4))}$$

where 'c' above a parameter represents its one's complement. Since not all places are connected to the input of transition tr_q, to represent connectivity from places to transitions we use the binary PTC matrix Q, with elements $q_{qk} \in \{0, 1\}$. Thus we find:

$$t_q(t+1) = \bigvee_{\forall k=1}^{n} (n_k^{\mathrm{c}}(t) \wedge q_{ik})^{\mathrm{c}} \wedge u\left[\left(\bigvee_{\forall k=1}^{n} (n_k^{\mathrm{c}}(t) \wedge q_{ik}) \right)^{\mathrm{c}} - \mathrm{th}_q \right] \qquad \text{(revised (4))}$$

For m transitions, as in Fig. 2, we will have m expressions as presented above. Combining all the m expressions we get the following state-space form for token updating at transitions:

$$T(t+1) = (Q \circ N^{\mathrm{c}}(t))^{\mathrm{c}} \wedge U[(Q \circ N^{\mathrm{c}}(t))^{\mathrm{c}} - \mathrm{Th}] \qquad (7)$$

where Th = threshold vector of dimension $(m \times 1)$, the ith component of which represents the threshold of transition tr_i, U = unit step vector of dimension $(m \times 1)$, and N^c = complement of the belief vector, the ith component of which is the complement of the ith component of N. The other parameters in expression (7) have the same meanings as defined earlier.

3.3. State-space model for weights

The weight updating equation given by expression (3) can be written in vector–matrix form as follows:

$$W(t+1) = (1-\alpha)W(t) + ((N(t) \cdot T^{\mathrm{T}}(t)) \wedge P) \tag{8}$$

where $W = [w_{ij}]$ is a weight matrix of dimension $(n \times m)$, P = binary TPC matrix of dimension $(n \times m)$ with element $p_{ij} \in \{0, 1\}$, where p_{ij} represents connectivity from tr_j to p_i, and 'T' over a vector denotes its transposition. The other parameters of expression (8) have been defined earlier.

4. Stability analysis of the cognitive model

The results of stability analysis are presented in Theorems 1–3.

Theorem 1. *For $n_k(0)$ lying between 0 and 1, for all k, $n_i(t)$ in expression (5) always converges.*

Proof. Since $n_k(0)$ for all k are bounded in $[0, 1]$, and expressions (4) and (5) involve AND (min) and OR (max) operators, $n_i(t)$ at any time t remains bounded in $[0, 1]$. Further, from expression (5) it is evident that $n_i(t+1) \geqslant n_i(t)$. Thus $n_i(t+1)$ can never decrease with respect to $n_i(t)$, and thus $n_i(t+1)$ cannot be oscillatory. Consequently, $n_i(t+1)$ being bounded in $[0, 1]$ and oscillation-free converges to a stable point after some finite number of iterations. □

Theorem 2. *The fuzzy truth tokens $t_j(t)$ at transition tr_j for all j given by expression (4) converge to stable points for $n_k(0)$ lying in $[0, 1]$ for all k.*

Proof. Since fuzzy beliefs at places in the cognitive net converge to stable points, therefore, by expression (4) the proof of the theorem is obvious. □

Theorem 3. *The weights in the encoding model given by expression (3) satisfy the following conditions:*

stable *when* $0 < \alpha < 2$,
limit cycle *when* $\alpha = 2$ *and*
unstable *when* $\alpha > 2$.

Proof. Rewriting expression (3) by extended difference operator E, we find

$$(E - 1 + \alpha)w_{ij}(t) = t_j(t) \cdot n_i(t) \tag{9}$$

Now, to prove the theorem we need to solve the above equation. The complementary function (CF) for the above equation is given by

$$\text{CF} : w_{ij}(t) = A(1 - \alpha)^t \tag{10}$$

where A is a constant.

The particular integral (PI) for expression (9) is given by

$$\text{PI} : w_{ij}(t) = \frac{t_j(t)n_i(t)}{E - 1 + \alpha} \tag{11}$$

Since the PI represents the steady-state value of $w_{ij}(t)$, let us consider the steady-state values of $t_j(t)$ and $n_i(t)$ in expression (8).

Let $t_j(t)$ as $t \to \infty$ be called t_j^* and $n_i(t)$ as $t \to \infty$ be called n_i^*.

On replacement of $t_j(t)$ and $n_i(t)$ by their steady-state values in expression (11), the numerator of expression (11) becomes a constant. Therefore, we set $E = 1$ in expression (11), which yields the PI given by

$$\text{PI} : w_{ij}(t) = \frac{t_j^* n_i^*}{\alpha} \tag{12}$$

The complete solution of Eq. (9) is then given by

$$w_{ij}(t) = A(1 - \alpha)^t + \frac{t_j^* n_i^*}{\alpha} \tag{13}$$

Substituting $t = 0$ in expression (13) we find the value of A, which on further substitution in (13) yields the complete solution given by

$$w_{ij}(t) = (w_{ij}(0) - t_j^* n_i^* / \alpha)(1 - \alpha)^t + \frac{t_j^* n_i^*}{\alpha} \tag{14}$$

The conditions for stability, limit cycles and instability, now, directly follow from expression (14). □

In the following example, we verify the conditions for stability, limit cycles and instability of the encoding process and unconditional stability of the recall process.

Example 2. Consider the cognitive map for bird (X) given in Fig. 4. The initial values of beliefs, weights and thresholds are shown in the figure. The encoding and recall models given in expressions (6)–(8) are repeated until limit cycle, steady-state or instability is observed on the plots (see Fig. 5). The results obtained in these figures are consistent with Theorem 3.

5. Computer simulation

The cognitive map of Fig. 6 is a graph, representing a system for automated collision/accident-free car driving. For this map, we initialize matrix $W(0)$, vector $N(0)$ and vector Th as follows:

W (0) =

From transition

To place	tr_1	tr_2	tr_3	tr_4	tr_5	tr_6	tr_7	tr_8
P_1	0	0	0	0	0	0	0	0
P_2	0	0	0	0	0	0	0	0
P_3	0	0	0	0	0	0	0	0
P_4	0.95	0	0.85	0	0	0.75	0	0
P_5	0	0.7	0	0	0	0	0	0
P_6	0	0	0	0	0	0	0	0
P_7	0	0	0	0	0	0	0	0
P_8	0	0	0	0.8	0.4	0	0	0
P_9	0	0	0	0	0	0	0	0
p_{10}	0	0	0	0	0	0	0	0
p_{11}	0	0	0	0	0	0	0.9	0
p_{12}	0	0	0	0	0	0	0	0.85

$$N(0) = [0.6 \quad 0.7 \quad 0.5 \quad 0.75 \quad 0.6 \quad 0.6 \quad 0.9 \quad 0.6 \quad 0.8 \quad 0.7 \quad 0.5 \quad 0.6]^{\mathrm{T}}$$

$$\mathrm{Th} = [0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1]^{\mathrm{T}}$$

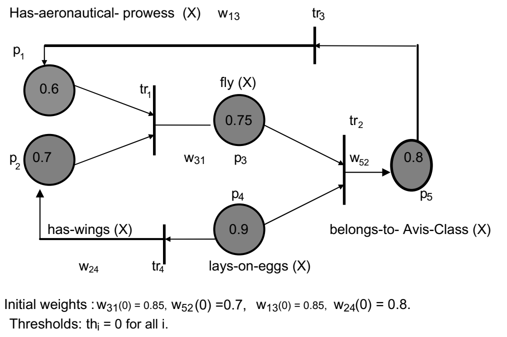


Fig. 4. A cognitive map for birds built with FPN.

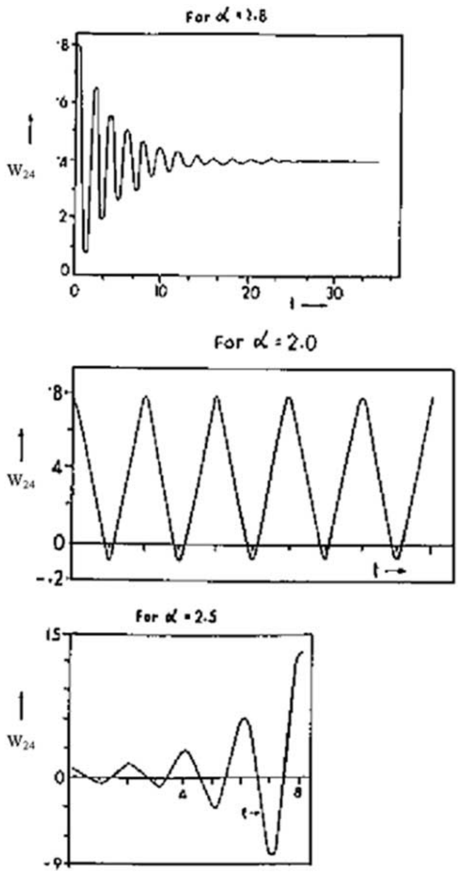


Fig. 5. Effect of α on the dynamics of a sample weight w_{24} in Fig. 4.

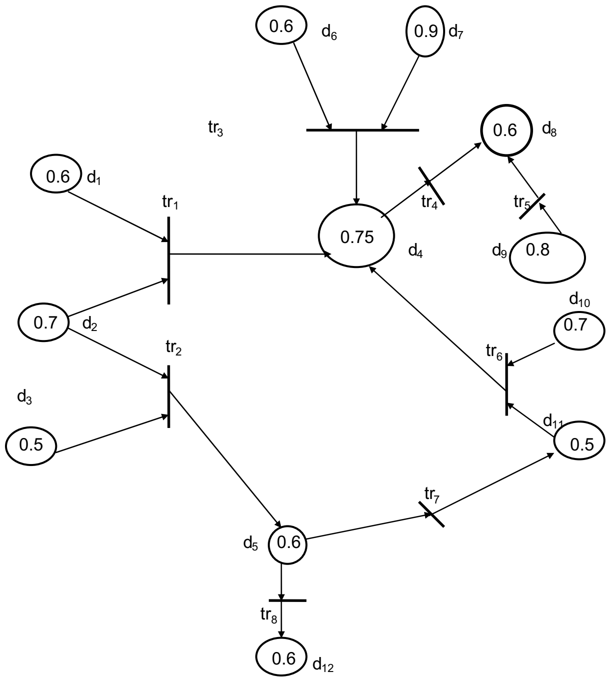
The P and Q matrices for the above system, as obtained from Fig. 6, are given below.

P = (To places / From transition)

	tr_1	tr_2	tr_3	tr_4	tr_5	tr_6	tr_7	tr_8
p_1	0	0	0	0	0	0	0	0
p_2	0	0	0	0	0	0	0	0
p_3	0	0	0	0	0	0	0	0
p_4	1	0	1	0	0	1	0	0
p_5	0	1	0	0	0	0	0	0
p_6	0	0	0	0	0	0	0	0
p_7	0	0	0	0	0	0	0	0
p_8	0	0	0	1	1	0	0	0
p_9	0	0	0	0	0	0	0	0
p_{10}	0	0	0	0	0	0	0	0
p_{11}	0	0	0	0	0	0	1	0
p_{12}	0	0	0	0	0	0	0	1

Q = (To transition / From place)

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}
tr_1	1	1	0	0	0	0	0	0	0	0	0	0
tr_2	0	1	1	0	0	0	0	0	0	0	0	0
tr_3	0	0	0	0	0	1	1	0	0	0	0	0
tr_4	0	0	0	1	0	0	0	0	0	0	0	0
tr_5	0	0	0	0	0	0	0	0	1	0	0	0
tr_6	0	0	0	0	0	0	0	0	0	1	1	0
tr_7	0	0	0	0	1	0	0	0	0	0	0	0
tr_8	0	0	0	0	1	0	0	0	0	0	0	0



d_1=Car behind side-car narrow in breadth, d_2 = Side-car of the front car too close, d_3 = Car behind side-car is wide, d_4 = Front car speed decreases, d_5= Front car speed increases, d_6 =Passer-by changes his/ her direction, d_7= Passer-by crosses the road, d_8= Rear car speed decreases, d_9 =Front car changes direction, d_{10}= Rear car changes direction, d_{11}= Rear car speed increases, d_{12}= Rear car keeps safe distance with respect to front car.

Fig. 6. A fuzzy cognitive map, representing a car driver's problem.

Eqs (8), (7) and (6) are updated, in that order, with $\alpha = 1.8$ (<2) until convergence in weights is attained. The steady-state values of weights, thus obtained, are saved by the cognitive system for application during the recognition phase. The steady state W^*, obtained through simulation, is given by

$$
\begin{aligned}
&w^*_{41} = 0.25, \quad w^*_{43} = 0.25, \quad w^*_{46} = 0.25, \quad w^*_{56} = 0.17, \quad w^*_{84} = 0.31, \\
&w^*_{85} = 0.33, \quad w^*_{11,7} = 0.2, \quad w^*_{12,8} = 0.2, \quad \text{and} \\
&w_{ij} = 0 \quad \text{for all other values of } i, j.
\end{aligned}
$$

Now, with a new $N(0)$ vector and the W^* matrix, the cognitive system can derive new steady-state inferences N^* by using expressions (7) and (6), in that order. The value of $N(0)$ and N^* are presented below:

$$N(0) = [0.2 \quad 0.3 \quad 0.4 \quad 0.0 \quad 0.0 \quad 0.3 \quad 0.35 \quad 0.0 \quad 0.4 \quad 0.3 \quad 0.0 \quad 0.0]^{\mathrm{T}}$$

$$N^{*} = [0.2 \quad 0.3 \quad 0.4 \quad 0.25 \quad 0.17 \quad 0.3 \quad 0.35 \quad 0.33 \quad 0.4 \quad 0.3 \quad 0.17 \quad 0.17]^{\mathrm{T}}$$

Among the concluding places $\{p_4, p_5, p_8, p_{11}, p_{12}\}$, p_8 has the highest steady-state belief ($=0.33$), and therefore d_8 (RC-SD) has to be executed.

6. Significance of the results

Stability analysis of a fuzzy cognitive map has been accomplished in detail. The result of the stability analysis shows that the encoding model is conditionally stable ($0 < \alpha < 2$), while the recall model is unconditionally stable. Further, the cognitive map never destroys any weights (connectivity) during the process of encoding. This is evident from the fact that when $w_{ij}(0) \neq 0$, w^{*}_{ij} too is non-zero for any i, j.

7. Knowledge refinement by Hebbian learning

This section presents a new method for automated estimation of certainty factors of knowledge from the proven and historical databases of a typical reasoning system. Here certainty factors have been modeled by weights in a special type of recurrent fuzzy neural Petri net. The beliefs of the propositions, collected from the historical databases, are mapped at places of a fuzzy Petri net and the weights of directed arcs from transitions to places are updated synchronously following the Hebbian learning principle until an equilibrium condition [20,21] is reached. The model for weight adaptation has been chosen for maintaining consistency among the initial beliefs of the propositions and thus the derived steady-state weights represent a more accurate measure of certainty factors than those assigned by a human expert.

7.1. The encoding model

The process of encoding of weights consists of three basic steps, presented below:

Step-I. A transition tr_i is enabled if all its input places possess tokens. An enabled transition is firable. On firing of a transition tr_i, its FTT t_i is updated using expression (15) [20], where places $p_k \in I(\mathrm{tr}_i)$, n_k is the belief of proposition mapped at place p_k, and th_i is the threshold of transition tr_i:

$$t_i(t+1) = \left(\bigwedge_{1 \leqslant k \leqslant n} n_k(t) \right) \wedge u\left[\left(\bigwedge_{1 \leqslant k \leqslant n} n_k(t) \right) - \mathrm{th}_i \right] \tag{15}$$

Expression (15) reveals that if,

$$t_i(t+1) = \begin{cases} \bigwedge_{1 \leqslant k \leqslant n} n_k(t) \bigwedge_{1 \leqslant k \leqslant n} n_k > \text{th}_i, \\ 0 \qquad \text{otherwise} \end{cases}$$

Step-II. After the FTTs at all the transitions are updated synchronously, we revise the fuzzy beliefs at all places concurrently. The fuzzy belief n_j at place p_j is updated using expression (16a) or (16b) (when p_i is an axiom, having no input arc):

$$n_i(t+1) = \begin{cases} \bigvee_{j=1}^{m} (t_j(t+1) \wedge w_{ij}(t)), \quad \text{when} \quad p_i \in O(\text{tr}_j) & \text{(16a)} \\ n_i(t), \quad \text{when} \quad p_i \text{ is an axiom} & \text{(16b)} \end{cases}$$

Step-III. Once the updating of fuzzy beliefs is over, the weights w_{ij} of the arc connected between transition tr_j and its output place p_i are updated following Hebbian learning [25]:

$$w_{ij}(t+1) = t_j(t+1) \wedge n_i(t+1) \tag{17}$$

The above three-step process for encoding is repeated until the weights do not change further. Such a time-invariant state is called an equilibrium state. The steady-state values of weights are saved for subsequent reasoning in analogous problems.

Theorem 4. *The encoding process of weights in a cognitive map realized with FPN is unconditionally stable.*

Proof. Since $n_k(0)$ for all k is bounded in $[0, 1]$, and the encoding model (expressions (15)–(17)) involves only AND (min) and OR (max) operators, $n_k(t)$ will remain bounded at any time t. Further, from expression (16a) we have

$$\begin{aligned} n_i(t+1) &= \bigvee_{j=1}^{m} (t_j(t+1) \wedge w_{ij}(t)) \quad \text{((16a) rewritten)} \\ &= \bigvee_{j=1}^{m} \{t_j(t+1) \wedge t_j(t) \wedge n_i(t)\} \quad \text{by (17)} \\ &= \left[\bigvee_{j=1}^{m} \{t_j(t+1) \wedge t_j(t)\} \right] \wedge n_i(t) \end{aligned}$$

Thus $n_i(t+1) \leqslant n_i(t)$. Consequently, $n_i(t+1)$ can only decrease with respect to its last value, and thus it cannot have an oscillation. Therefore, $n_i(t+1)$ being bounded and oscillation-free is stable. This holds for $k =$ any i, for $1 \leqslant i \leqslant n$, where n is the number of places in the FPN.

If $n_k(t+1)$ for all k is stable, then by Eq. (15), $t_j(t+1)$ also is stable. Finally, $n_i(t+1)$ and $t_j(t+1)$ being stable, $w_{ij}(t+1)$ by expression (17) is also stable. Hence, the encoding process of weights in a cognitive map by the proposed model is stable. □

7.2. The recall/reasoning model

The reasoning models of a recurrent FPN have been reported elsewhere [9–14]. During the reasoning phase, we can use any of these models including the new model proposed below.

The reasoning/recall model in an FPN can be carried out in the same way as in the first two steps of the encoding model, with the following exceptions:

- While initiating the reasoning process, the known fuzzy beliefs for the propositions of a problem are to be assigned to the appropriate places. It is to be noted that in the encoding model the fuzzy beliefs of propositions were submitted using proven case histories.
- The reasoning model should terminate when the fuzzy beliefs associated with all propositions converge to stable points, i.e., when for all places, $|n_i(t^*+1) - n_i(t^*)| < \epsilon$ for $t = \min(t^*)$, where ϵ is a positive number, however small.

The fuzzy beliefs thus obtained after convergence are used to interpret the results of typical analogous reasoning problems.

The execution of the reasoning model is referred to as the belief revision cycle [9].

Theorem 5. *The recall process in an FPN unconditionally converges to stable points in belief space.*

Proof. Directly follows from the first part of the proof of Theorem 4 showing that $n_i(t+1)$ for all i is stable. □

7.3. Case study by computer simulation

In this study, we consider two proven case histories described by (Rule base I, Database I) and (Rule base II, Database II). The beliefs of each proposition in the FPNs (Figs. 7 and 8) for these two case histories are known. The encoding model for the cognitive map has been used to estimate the certainty factors (CFs) of the causal rules in either case. In case the two estimated CFs of a rule obtained from the two case-histories differ, we take the average of the two values as its CF [19].

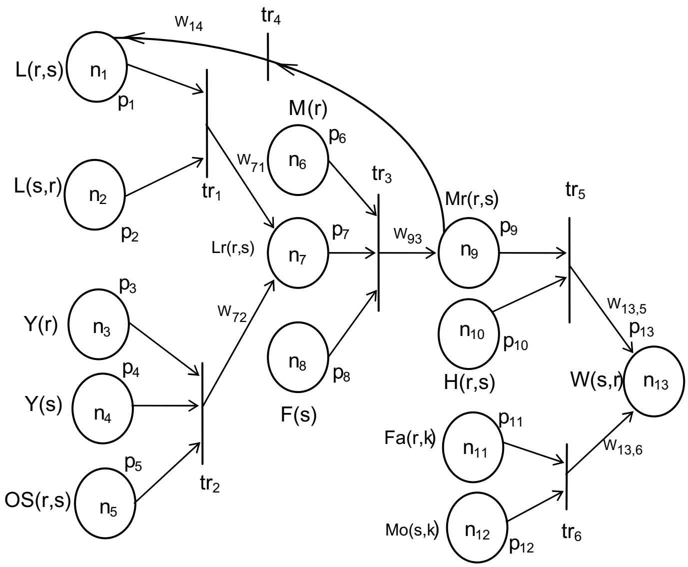


Fig. 7. An FPN with initially assigned known beliefs and random weights.

Case History I

Rule base I:

PR1: Loves(x,y), Loves(y,x) $\rightarrow$ Lover(x,y)
PR2: Young(x), Young(y), Opposite-Sex(x,y) $\rightarrow$ Lover(x,y)
PR3: Lover(x,y), Male(x), Female(y) $\rightarrow$ Marries(x,y)
PR4: Marries(x,y) $\rightarrow$ Loves(x,y)
PR5: Marries(x,y), Husband(x,y) $\rightarrow$ Wife(y,x)
PR6: Father(x,z), Mother(y,z) $\rightarrow$ Wife(y,x)

Database I:

Loves(ram, sita), Loves(sita, ram), Young(ram),
Young(sita), Opposite-Sex(ram, sita),
Male(ram), Female(sita), Husband(ram, sita),
Father(ram, kush), Mother (sita, kush)

While reasoning in analogous problems, the rules may be assigned with the CFs estimated from the known histories. In this example, case-III is a new test problem, whose knowledge base is a subset of the union of the knowledge bases of the first two cases. Thus the CFs of the rules are known. In case-III, only the

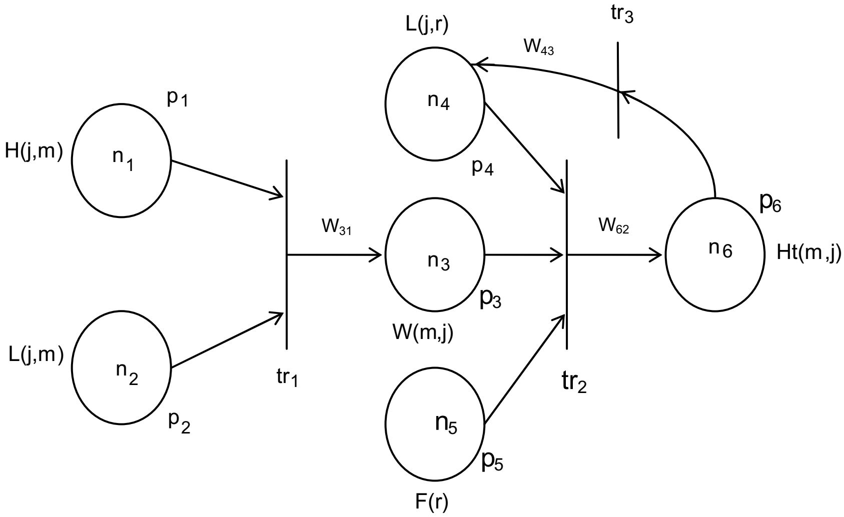


Fig. 8. A second FPN with known initial beliefs and random weights.

initial beliefs of the axioms are assumed to be known. The aim is to estimate the steady-state belief of all propositions in the network. Since stability of the reasoning model is guaranteed, the belief revision process is continued until steady-state is reached. In the present simulation, steady-state occurs in the reasoning model of case-III after five belief revision cycles. Once the steady-state condition is reached, the network may be used for generating new inferences.

The FPN in Fig. 7 has been created using rule base I and database I from a typical case history. The fuzzy beliefs of the places in Fig. 7 are found from proven historical database. The initial weights in the network are assigned arbitrarily and the steady-state values of weights (see Table 1) are computed by using the encoding model.

Case History II

Rule base II:

PR1: Wife(y, x), Loves(x, z), Female(z) $\rightarrow$ Hates(y, x)
PR2: Husband(x, y), Loves(x, y) $\rightarrow$ Wife(y, x)
PR3: Hates(z, x) $\rightarrow$ Loves(x, y)

Database II:

Husband(jadu, mala), Loves(jadu, mala),
Loves(jadu, rita), Female(rita)

Table 1
Parameters of Case History I

Initial weights w_{ij}	$w_{71} = 0.8$, $w_{72} = 0.7$, $w_{93} = 0.6$, $w_{14} = 0.9$, $w_{13,5} = 0.8$, $w_{13,6} = 0.5$
Initial fuzzy beliefs n_i	$n_1 = 0.2$, $n_2 = 0.8$, $n_3 = 0.75$, $n_4 = 0.9$, $n_5 = 0.6$, $n_6 = 0.75$, $n_7 = 0.35$, $n_8 = 0.85$, $n_9 = 0.45$, $n_{10} = 0.85$, $n_{11} = 0.7$, $n_{12} = 0.65$, $n_{13} = 0.3$
Steady-state weights after 4 iterations	$w_{71} = 0.35$, $w_{72} = 0.60$, $w_{93} = 0.35$, $w_{14} = 0.35$, $w_{13,5} = 0.35$, $w_{13,6} = 0.50$
$\text{th}_j = 0$ for all transitions tr_j	

Table 2
Parameters of Case History II

Initial weights w_{ij}	$w_{31} = 0.75$, $w_{62} = 0.95$, $w_{43} = 0.8$
Initial fuzzy beliefs n_i	$n_1 = 0.8$, $n_2 = 0.7$, $n_3 = 0.1$, $n_4 = 0.2$, $n_5 = 0.9$, $n_6 = 0.3$
Steady-state weights after 3 iterations	$w_{31} = 0.7$, $w_{62} = 0.10$, $w_{43} = 0.10$
$\text{th}_j = 0$ for all transitions tr_j	

The FPN of Fig. 8 is formed with rule base II and database II. The system parameters of this FPN are presented in Table 2.

The Current Reasoning Problem

To solve a typical reasoning problem with given knowledge and databases (see below), we need to assign the derived weights from the last two case histories. The reasoning model can be used in this example to compute the steady-state belief of the proposition Hates (lata, ashoke) with the given initial beliefs of all the propositions. Table 3 shows the system parameters of the FPN in Fig. 9.

Current Rule base:

PR1: Loves(x, y), Loves(y, x) $\rightarrow$ Lover(x, y)
PR2: Young(x), Young(y), OS(x, y) $\rightarrow$ Lover(x, y)
PR3: Lover(x, y), Male(x), Female(y) $\rightarrow$ Marries(x, y)
PR4: Marries(x, y) $\rightarrow$ Loves(x, y)
PR5: Marries(x, y), Husband(x, y) $\rightarrow$ Wife(y, x)
PR6: Father(x, z), Mother(y, z) $\rightarrow$ Wife(y, x)

Table 3
Parameters of the current reasoning problem

Initial weight w_{ij} taken from the steady state CFs of corresponding rules from earlier Case Histories I and II shown in parenthesis	$w_{71} = w_{71}(\mathrm{I}) = 0.35$, $w_{72} = w_{72}(\mathrm{I}) = 0.60$, $w_{93} = w_{93}(\mathrm{I}) = 0.35$, $w_{14=}w_{14}(\mathrm{I}) = 0.35$, $w_{13,5} = w_{13,5}(\mathrm{I}) = 0.35$, $w_{13,6} = w_{13,6}(\mathrm{I}) = 0.50$, $w_{16,7} = w_{62}(\mathrm{II}) = 0.10$, $w_{15,8} = w_{43}(\mathrm{II}) = 0.10$
Initial fuzzy beliefs n_i	$n_1 = 0.4$, $n_2 = 0.8$, $n_3 = 0.75$, $n_4 = 0.85$, $n_5 = 0.65$, $n_6 = 0.9$, $n_7 = 0.3$, $n_8 = 0.7$, $n_9 = 0.3$, $n_{10} = 0.95$, $n_{11} = 0.65$, $n_{12} = 0.6$, $n_{13} = 0.25$, $n_{14} = 0.55$, $n_{15} = 0.35$, $n_{16} = 0.40$
Steady-state belief at place $\mathbf{p_{16}}$ for proposition Hates(l,a)	$\mathbf{n_{16} = 0.10}$
$\mathrm{th}_j = 0$ for all transitions tr_j	

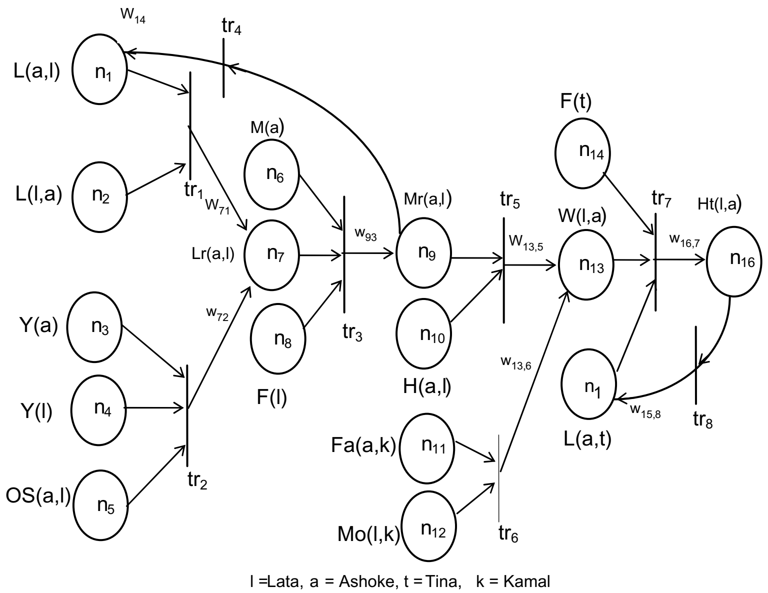


Fig. 9. An FPN used for estimating the belief of Ht(l,a) with known initial belief and CFs.

PR7: Wife(y, x), Loves(x, z), Female$(z) \rightarrow$ Hates(y, x)
PR8: Hates$(z, x) \rightarrow$ Loves(x, y)

Current Database:

Loves(ashoke, lata), Loves(lata, ashoke),

Young(ashoke), Young(lata), Opposite-Sex(ashoke, lata),
Male(ashoke), Female(lata), Husband(ashoke, lata),
Father(ashoke, kamal), Mother(lata, kamal),
Loves(ashoke,tina), Female(tina)

7.4. Significance of the results

The analysis of stability in Section 7 shows that both the encoding and the recall model of the fuzzy cognitive map are unconditionally stable. The time required for convergence of the proposed model is proportional to the number of transitions on the largest path (cascaded set of arcs) [11] in the network. The model could be used for determination of certainty factor of rules in a knowledge base by maintaining consistency among the beliefs of the propositions of known case histories.

8. Conclusions

The paper developed two novel models of fuzzy cognitive maps that are free from the limitations of the existing models. The first model employed Hebbian learning with an additional forgetfulness factor α. The conditional convergence of the first model in the encoding phase and unconditional convergence in the recall phase have been proved. The second model has been proved to converge to stable points in both encoding and recall phases.

Both the models can be utilized for knowledge acquisition/refinement from multiple experts. If the first model is used for knowledge refinement, then the FPNs representing case histories should have the same α. Selection of α is an important issue. A large α (approaching 2.0) causes an over-damped behavior of weights. On the other hand, a very small α (<0.2) requires significant time for the oscillations to settle down to stable values. Typically, α is chosen in the range $[1.0, 1.2]$ for stable learning without quick memory refresh.

The second model (the one that does not include α) can easily be employed for knowledge refinement from multiple experts. The fuzzy beliefs of the inferences derived by the second model during the recall phase, however, are less accurate in comparison to those of the first model. In brief, the first model provides a more accurate representation of cognitive learning by human beings, and naturally, its dynamics is more complex when compared to the second model.

One interesting observation about the first model is that for non-zero initial values of w_{ij}, w^*_{ij} too is non-zero. The significance of this lies in the fact that the model never destroys the structural topology of the cognitive map. This particular characteristic of the model ensures reliable fusion of knowledge from multiple experts without destroying any connectivity in the process of natural decay of weights.

Acknowledgements

We are grateful to the anonymous referees for their comments.

References

[1] A.J. Bugarin, S. Barro, Fuzzy reasoning supported by Petri nets, IEEE Trans. Fuzzy Syst. 2 (2) (1994) 135–150.
[2] T. Cao, A.C. Sanderson, Task sequence planing using Fuzzy Petri nets, IEEE Trans. Syst. Man Cybernet. 25 (5) (1995) 755–769.
[3] T. Cao, A.C. Sanderson, A fuzzy Petri net approach to reasoning about uncertainty in Robotic systems, in: Proc. IEEE Int. Conf. Robotics and Automation, Atlanta, GA, May 1993, pp. 317–322.
[4] T. Cao, Variable reasoning and analysis about uncertainty with fuzzy Petri nets, in: M.A. Marson (Ed.), Lecture Notes in Computer Science, vol. 691, Springer-Verlag, New York, 1993, pp. 126–145.
[5] J. Cardoso, R. Valette, D. Dubois, Petri nets with uncertain markings, in: G. Rozenberg (Ed.), Advances in Petri Nets, Lecture Notes in Computer Science, vol. 483, Springer-Verlag, New York, 1990, pp. 65–78.
[6] S.-M. Chen, J.-S. Ke, J.-F. Chang, Knowledge representation using fuzzy Petri nets, IEEE Trans. Knowledge Data Eng. 2 (3) (1990) 311–319.
[7] A. Daltrini, Modeling and Knowledge processing based on the extended fuzzy Petri nets, M.Sc. degree thesis, UNICAMP-FEE0DCA, May 1993.
[8] M.L. Garg, S.I. Ashon, P.V. Gupta, A fuzzy petri NET for knowledge representation and reasoning, Inform. Process. Lett. 39 (1991) 165–171.
[9] A. Konar, Uncertainty management in expert system using fuzzy Petri nets, Ph.D. dissertation, Jadavpur University, India, 1994.
[10] A. Konar, Artificial Intelligence and Soft Computing: Behavioral and Cognitive Modeling of the Human Brain, CRC Press, Boca Raton, 1999.
[11] A. Konar, A.K. Mandal, Uncertainty management in expert systems using fuzzy Petri nets, IEEE Trans. Knowledge Data Eng. 8 (1) (1996) 96–105.
[12] A. Konar, A.K. Mandal, Stability analysis of a non-monotonic Petri Nets for diagnostic systems using fuzzy logic, in: 33rd Midwest Symp. on Circuits, Systems and Computers, Canada, 1991.
[13] A. Konar, A.K. Mandal, Non-monotonic Reasoning in Expert Systems using Fuzzy Logic, AMSE Publication, 1992.
[14] S. Konar, A. Konar, A.K. Mandal, Analysis of fuzzy Petri net models for reasoning with inexact data and knowledge-base, in: Proc. of Int. Conf. on Control, Automation, Robotics and Computer Vision, Singapore, 1994.
[15] B. Kosko, Fuzzy cognitive maps, Int. J. Man–Machine Stud. 24 (1986) 65–75.
[16] B. Kosko, Fuzzy Engineering, Prentice-Hall, 1997, Chapter 15, pp. 499–527.
[17] H.P. Lipp, G. Gunther, A fuzzy Petri net concept for complex decision making process in production control, in: Proc. First European Congress on Fuzzy and Intelligent Technology (EUFIT '93), Aachen, Germany, vol. I, pp. 290–294.

[18] C.G. Looney, Fuzzy Petri nets for rule-based decision making, IEEE Trans. Syst. Man Cybernet. 18 (1) (1988) 178–183.
[19] S. Pal, A. Konar, Cognitive reasoning using fuzzy neural nets, IEEE Trans. Syst. Man Cybernet. B 26 (4) (1996).
[20] B. Paul, A. Konar, A.K. Mandal, Estimation of certainty factor of rules by associative memory realised with fuzzy Petri nets, in: Proc. IEEE Int'l Conf. on Automation, Robotics and Computer Vision, Singapore, 1996.
[21] B. Paul, A. Konar, A.K. Mandal, Estimation of certainty factor of knowledge with Fuzzy Petri nets, presented in FUZZ-IEEE '98 Int'l Conf, Alaska, 1998.
[22] W. Pedrycz, F. Gomide, A generalized fuzzy Petri net model, IEEE Trans. Fuzzy Syst. 2 (4) (1994) 295–301.
[23] W. Pedrycz, Fuzzy sets Engineering, CRC Press, Boca Raton, 1995.
[24] J.F. Peters, N. Sohi, Coordination of multi-Agent systems with fuzzy clocks, Concurrent Engineering Research and Applications IV (1) (1996).
[25] H. Scarpelli, F. Gomide, High level Fuzzy Petri nets and backward reasoning, in: Bouchon-Meunier, Yager, Zadeh (Eds.), Fuzzy Logic and Soft Computing, World Scientific, 1995.
[26] H. Scarpelli, F. Gomide, R. Yager, A reasoning algorithm for high level Fuzzy Petri nets, IEEE Trans. Fuzzy Syst. 4 (3) (1996) 282–295.
[27] P.K. Simpson, Artificial Neural System: Foundation, Paradigm, Application and Implementation, Pergamon Press, 1990.