Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

A Vague Reasoning Algorithm for Rule-Based Systems Based on Extended Fuzzy Petri Nets

Shyi-Ming Chen and Yuh-Shin Shiau

Department of Computer and Information Science
National Chiao Tung University, Hsinchu, Taiwan, R. O. C.

Abstract

In this paper, we present a vague reasoning algorithm for rule-based systems based on extended fuzzy petri nets. The proposed algorithm is more flexible than the one we presented in [7] due to the fact that it allows fuzzy IF-THEN rules and fuzzy IF-THEN-ELSE rules to appear in the knowledge base of a rule-based system and it allows the truth values of the propositions appearing in the rules to be represented by vague values in [0, 1] rather than real values between zero and one.

Keywords: Extended Fuzzy Petri Net, Fuzzy Petri Net, Fuzzy Production Rule, Knowledge Representation, Rule-Based System, Vague Reasoning, Vague Value.

1. Introduction

Since fuzzy Petri nets provide an effective approach for representing fuzzy knowledge, the applications of fuzzy Petri nets have been investigated by many researchers. In [1], Bugarin et al. developed a representation model for fuzzy reasoning supported by Petri nets. In [2], Cao et al. presented a method for task sequence planning using fuzzy Petri nets. In [8], Garg et al. presented a fuzzy Petri net model for representing knowledge and presented an algorithm for checking the consistency of a fuzzy knowledge base via a set of reduction rules. In [13], Konar et al. developed new techniques for uncertainty management in expert systems using fuzzy Petri nets. In [14], Looney presented a fuzzy Petri net model for rule-based decisionmaking. In [15], Looney proposed a fuzzy Petri net algorithm and investigated a fuzzy logic rule-based control application based on fuzzy Petri nets, where a fuzzy Petri net train controller is investigated. In [17], Pedrycz et al. proposed a generalized fuzzy Petri net model based on the use of logic based neurons, where the learning aspects associated with the nets are investigated. In [18], Yu presented a fuzzy Pr/T net-system model (FPM) for knowledge representation and processing of the fuzzy production rules in knowledge-based systems, where a fuzzy reasoning algorithm for FPM is also proposed. In [7], we have presented a fuzzy Petri net model (FPN) to represent the fuzzy production rules of a rule-based system and presented an algorithm to perform fuzzy reasoning based on the fuzzy Petri net model, where the truth value of each proposition is represented by a real value between zero and one. However, this single value combines the degree of truth and the degree of false of the proposition. Furthermore, the fuzzy production rules used in [7] are restricted to fuzzy IF-THEN rules. If we can allow fuzzy IF-THEN-ELSE rules to be used for knowledge representation and allow the truth values of the propositions appearing in the rules to be represented by vague values [9] in [0, 1] rather than real values between zero and one, then there is room for more flexibility. According to [9], a vague value x is represented by $[t_x, 1 - f_x]$, where t_x indicates the degree of truth, f_x indicates the degree of false, $1 - t_x - f_x$ indicates the unknown part, $0 \leq t_x \leq 1 - f_x \leq 1$, and $t_x + f_x \leq 1$.

In [5], we have presented vague reasoning techniques for rule-based systems. In [6], we have presented an extended fuzzy Petri net model (EFPN) to model the fuzzy production rules of a rule-based system. In this paper, we extend the works of [5], [6], and [7] to propose an efficient algorithm for performing vague reasoning automatically. The proposed vague reasoning algorithm can determine whether there exists an antecedent-consequence relationship between proposition d_s to proposition d_j. Furthermore, if the vague truth value of proposition d_s is given, then the vague truth value of proposition d_j can be evaluated by the proposed algorithm. The proposed algorithm is more flexible than the one presented in [7] due to the fact that it allows fuzzy IF-THEN rules and fuzzy IF-THEN-ELSE rules to be used for knowledge representation and it allows the truth values of the propositions appearing in the rules to be represented by vague values in [0, 1] rather than real values between zero and one. This vague reasoning capability allows the computers to perform reasoning in a more flexible and more intelligent manner.

The rest of the paper is organized as follows. In Section 2, we briefly review the extended fuzzy Petri net model (EFPN) and the vague reasoning techniques from [5] and [6]. In Section 3, we present a vague reasoning algorithm for rule-based systems based on the extended fuzzy Petri net model. In Section 4, we use an example to illustrate the vague reasoning process. The conclusions are discussed in Section 5.

2. Extended Fuzzy Petri Nets and Vague Reasoning Techniques

In this section, we briefly review the vague reasoning techniques from [5] and the extended fuzzy Petri net model (EFPN) from [6]. An extended fuzzy Petri net is a bipartite directed graph which contains two types of nodes: places and transitions, where circles represent places, and bars represent transitions. Each place may or may not contain a token associated with a vague truth value in [0, 1]. Each transition is associated with a certainty factor value between zero and one. The relationships from places to transitions and from transitions to places are represented by directed arcs. There are two kinds of directed arcs from transitions to places, i.e., the positive arcs, denoted by "$\rightarrow$", and the negative arcs, denoted by "$\circ\!\!\rightarrow$". A generalized extended fuzzy Petri net structure can be defined as an 8-tuple:

$$\text{EFPN} = (\text{P, T, D, I, O, f}, \delta, \beta),$$

where

$P = \{p_1, p_2, \ldots, p_n\}$ is a finite set of places,

$T = \{T_1, T_2, \ldots, T_m\}$ is a finite set of transitions,

$D = \{d_1, d_2, \ldots, d_n\}$ is a finite set of propositions,

$P \cap T \cap D = \varnothing$, $|P| = |D|$,

I: $T \rightarrow P^{\infty}$ is the input function, a mapping from transitions to bags of places,

O: $T \rightarrow P^{\infty}$ is the output function, a mapping from transitions to bags of places,

f: $T \rightarrow [0, 1]$ is an association function, a mapping from transitions to real values between zero and one,

δ: $P \rightarrow [0, 1]$ is an association function, a mapping from places to vague values in [0, 1].

β: $P \rightarrow D$ is an association function, a bijective mapping from places to propositions.

An extended fuzzy Petri net with some places containing tokens is called a marked extended fuzzy Petri net. In a marked extended fuzzy Petri net, the token in a place p_i is represented by a labeled dot $\overset{\delta(p_i)}{\bullet}$. The token value in a place p_i, $p_i \in P$, is denoted by $\delta(p_i)$, where $\delta(p_i) = [t_i, 1 - f_i]$, $0 \le t_i \le 1 - f_i \le 1$, and $t_i + f_i \le 1$. If $\delta(p_i) = [t_i, 1 - f_i]$ and $\beta(p_i) = d_i$, then it indicates that the degree of truth and the degree of false of proposition d_i are t_i and f_i, respectively.

By using an extended fuzzy Petri net, the fuzzy production rule

$$R_i\text{: IF } d_j \text{ THEN } d_k \text{ ELSE } d_w \text{ (CF} = \mu_i)$$

can be modeled as shown in Fig. 1.

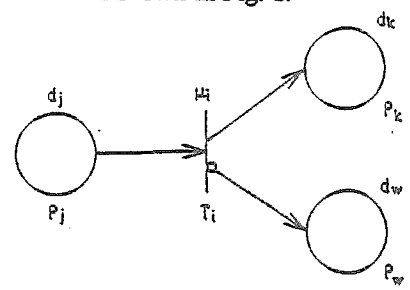

Fig. 1. An extended fuzzy Petri net.

In an extended fuzzy Petri net, a transition T_i may be enabled to fire. A transition T_i is enabled if there is a token in each of its input places. A transition T_i fires by removing the tokens from its input places and then depositing one token into each of its output places. Firing fuzzy production rules can be considered as firing transitions. For example, assume that the vague truth value of the proposition d_j of the above fuzzy production rule is $[t_j, 1 - f_j]$, then the vague reasoning process of the above rule can be modeled by a marked extended fuzzy Petri net as shown in Fig. 2.

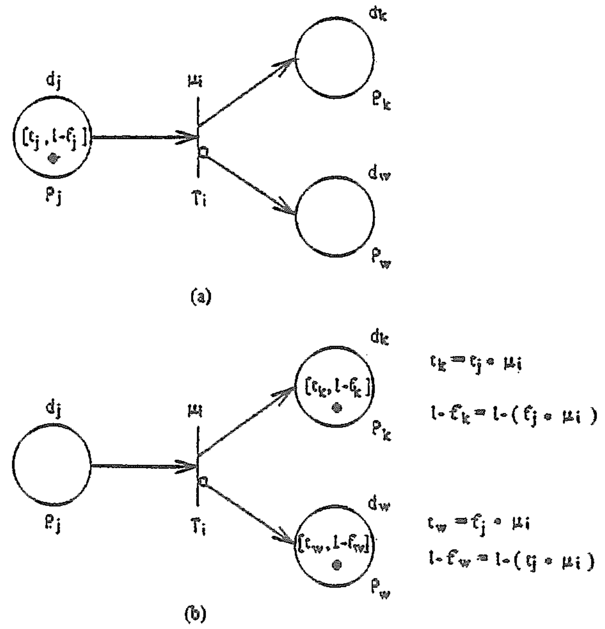

Fig. 2. Firing a marked extended fuzzy Petri net. (a) Before firing transition T_i. (b) After firing transition T_i.

Let T_a be a transition, and p_i, p_j, and p_k be three places. If $p_i \in I(T_a)$ and $p_k \in O(T_a)$, then p_k is called immediately reachable [5] from p_i. For example, assume that the fuzzy production rule

R_1: IF d_1 THEN d_2 ELSE d_3 (CF = μ)

has been modeled by an extended fuzzy Petri net as shown in Fig. 3. From Fig. 3, we can see that p_2 and p_3 are immediately reachable from p_1. In this case, p_2 is also called immediately direct reachable from p_1, and p_3 is also called immediately indirect reachable from p_1.

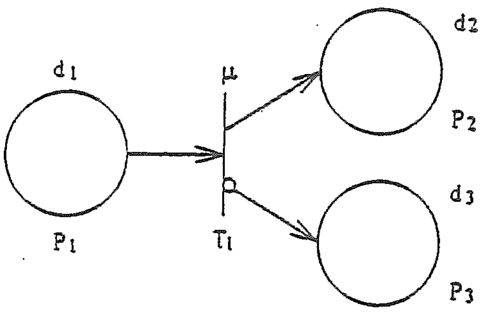

Fig. 3. An extended fuzzy Petri net.

If place p_k is immediately reachable from place p_i and place p_j is immediately reachable from place p_k, then place p_j is called reachable [7] from place p_i. The reachability relationship is the reflexive transitive closure of the immediate reachable relationship. The set of places which is immediate reachable from a place p_i is called the immediately reachability set [7] of p_i and is denoted by IRS(p_i). The set of places which is reachable from a place p_i is called the reachability set [7] of p_i and is denoted by RS(p_i). The set of places which is immediately direct reachable from a place p_i is called the immediately direct reachability set of p_i and is denoted by IDRS(p_i). The set of places which is immediately indirect reachable from a place p_i is called the immediately indirect reachable set of p_i and is denoted by IIRS(p_i). It is obvious that $\forall p_i \in P$,

$$\text{IRS}(p_i) = \text{IDRS}(p_i) \cup \text{IIRS}(p_i), \quad (15)$$

where "$\cup$" is the union operator between sets.

The reachability set RS(p_i), the immediate reachability set IRS(p_i), the immediate direct reachability set IDRS(p_i), and the immediate indirect reachability set IIRS(p_i) for each place p_i, $p_i \in P$, in Fig. 4 are shown in Table 1. From Table 1, we can see that IDRS(p_1) = {p_2}, IIRS(p_1) = {p_3}, IRS(p_1) = {p_2, p_3}, and RS(p_1) = (p_2, p_3, p_4, p_5, p_6, p_7}. They indicate that the place p_2 is immediate direct reachable from the place p_1, the place p_3 is immediate indirect reachable from the place p_1, the place p_2 and p_3 are immediate reachable from the place p_1, and the places p_2, p_3, p_4, p_5, p_6, and p_7 are reachable from the place p_1.

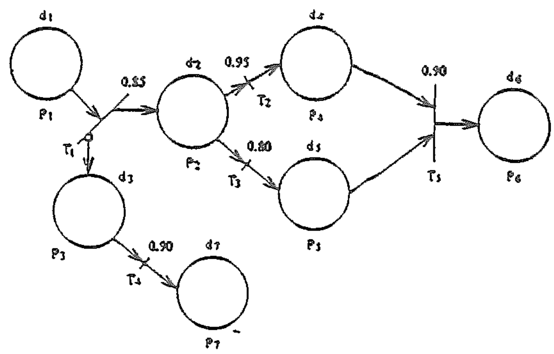

Fig. 4. An example of extended fuzzy Petri net.

TABLE 1
Immediate Direct Reachability Set, Immediate Indirect Reachability Set, Immediate Reachability Set, and The Reachability Set for Each Place p_i in Fig. 4

Place p_i	IDRS(p_i)	IIRS(p_i)	IRS(p_i)	RS(p_i)
p_1	{p_2}	{p_3}	{p_2, p_3}	{p_2, p_3, p_4, p_5, p_6, p_7}
p_2	{p_4, p_5}	ϕ	{p_4, p_5}	{p_4, p_5, p_6}
p_3	{p_7}	ϕ	{p_7}	{p_7}
p_4	{p_6}	ϕ	{p_6}	{p_6}
p_5	{p_6}	ϕ	{p_6}	{p_6}
p_6	{p_6}	ϕ	ϕ	ϕ
p_7	ϕ	ϕ	ϕ	ϕ

Let T_a be a transition, p_i and p_k be places. If $p_i \in I(T_a)$ and $p_k \in I(T_a)$, then p_i and p_k are called adjacent places [7] with respect to T_a. For example, from Fig. 4, we can see that p_4 and p_5 are adjacent places with respect to the transition T_5.

Let CF_{ij} denote the certainty factor value associated with the transition between places p_i and p_j, and let AP_{ij} denote a set of adjacent places of p_i, where $p_j \in$ IRS(p_i). The certainty factor table and the adjacent places table for Fig. 4 are shown in Table 2 and Table 3, respectively.

TABLE 2
Certainty Factor Table for Fig. 4

Place p_i	Place p_j	CF_{ij}
p_1	p_2	0.85
p_1	p_3	0.85
p_2	p_4	0.95
p_2	p_5	0.80
p_3	p_7	0.90
p_4	p_6	0.90
p_5	p_6	0.90

TABLE 3
Adjacent Places Table for Fig. 4

Place p_i	Place p_j	AP_{ij}
p_1	p_2	ϕ
p_1	p_3	ϕ
p_2	p_4	ϕ
p_2	p_5	ϕ
p_3	p_7	ϕ
p_4	p_6	$\{p_5\}$
p_5	p_6	$\{p_4\}$

In the next section, we will develop a vague reasoning algorithm based on [5], [6], and [7].

3. A Vague Reasoning Algorithm

In this section, we extend the works of [5], [6], and [7] to present a vague reasoning algorithm based on the extended fuzzy Petri net model (EFPN), where fuzzy IF-THEN rules and fuzzy IF-THEN-ELSE rules are used for knowledge representation, and the truth values of the propositions are represented by vague values in [0, 1]. The vague reasoning algorithm presented in the paper is an interactive algorithm. It can determine whether there exists an antecedent-consequence relationship from proposition d_s to proposition d_j.Furthermore, given the vague truth value of proposition d_s, the algorithm can evaluate the vague truth value of proposition d_j automatically. Assume that the vague truth value of proposition d_s given by the user is $[t_s, 1-f_s]$, where $0 \le t_s \le 1-f_s \le 1$ and $t_s + f_s \le 1$, and assume that he wants to ask what vague truth value proposition d_j might have, then because of $\beta(p_s) = d_s$ and $\beta(p_j) = d_j$, place p_s and place p_j are associated with the propositions d_s and d_j, respectively. In this case, the places p_s and p_j are called the starting place and the goal place, respectively. Furthermore, because the vague truth value of proposition d_s given by the user is $[t_s, 1-f_s]$, the token value in the place p_s is $[t_s, 1-f_s]$, i.e., $\delta(p_s) = [t_s, 1-f_s]$.

The vague reasoning algorithm proposed in this paper can automatically generate all the reasoning paths from a starting place p_s to a goal place p_j, and if the token value in the starting place p_s is known, then the token value in the goal place p_j can be evaluated. This implies that, if the vague truth value of proposition d_s is known, then the vague truth value of proposition d_j can be evaluated by the algorithm. The vague reasoning algorithm can be expressed by a tree. Each node in the tree is denoted by (p_k, $\delta(p_k)$, IRS(p_k), IDRS(p_k), IIRS(p_k)), where $p_k \in$ P. Let CF_{xy} denote the certainty value associated with a transition between place p_x and place p_y and let AP_{xy} denote a set of adjacent places of p_x, where $p_y \in$ IRS(p_x). The algorithm is now presented as follows:

<u>Vague Reasoning Algorithm:</u>

INPUT: the vague truth value $[t_s, 1-f_s]$ of proposition d_s, where $0 \le t_s \le 1-f_s \le 1$, $t_s + f_s \le 1$, t_s denotes the degree of truth of proposition d_s, and f_s denotes the degree of false of proposition d_s.

OUTPUT:the vague truth value $[t_j, 1-f_j]$ of proposition d_j, where $0 \le t_j \le 1-f_j \le 1$, $t_j + f_j \le 1$, t_j denotes the degree of truth of proposition d_j, and f_j denotes the degree of false of proposition d_j.

Step 1: Initially, the root node (p_s, $\delta(p_s)$, IRS(p_s), IDRS(p_s), IIRS(p_s)) is a nonterminal node, where
i) p_s is the starting place, and $\beta(p_s) = d_s$.
ii) $\delta(p_s) = [t_s, 1-f_s]$, and $[t_s, 1-f_s]$ is the vague truth value of proposition d_s given by the user, where $0 \le t_s \le 1-f_s \le 1$, $t_s + f_s \le 1$, t_s denotes the degree of truth of proposition d_s, and f_s denotes the degree of false of proposition d_s.
iii) IRS(p_s) is the immediate reachability set of the starting place p_s.
iv) IDRS(p_s) is the immediate direct reachability set of the starting place p_s.
v) IIRS(p_s) is the immediate indirect reachability set of the starting place p_s.

Step 2: Select one nonterminal node (p_i, $\delta(p_i)$, IRS(p_i), IDRS(p_i), IIRS(p_i)), where $\delta(p_i) = [t_i, 1-f_i]$, $0 \le t_i \le 1-f_i \le 1$, $t_i + f_i \le 1$. If IRS(p_i) = ϕ or for all $p_k \in$ IRS(p_i), the goal place $p_j \notin$ RS(p_k), then mark the node as a terminal node.
If the goal place $p_j \in$ IRS(p_i) and $CF_{ij} = \mu$, where $\mu \in [0, 1]$, then
if $p_j \in$ IDRS(p_i), then create a new node (p_j, $\delta(p_j)$, IRS(p_j), IDRS(p_j), IIRS(p_j)) in the tree, and an arc, labeled μ, is directed from the node ((p_i, $\delta(p_i)$, IRS(p_i), IDRS(p_i), IIRS(p_i)) to the node (p_j, $\delta(p_j)$, IRS(p_j), IDRS(p_j), IIRS(p_j)),

where $\delta(p_j) = [t_i * \mu, 1 - (f_i * \mu)]$. In this case, the node $((p_j, \delta(p_j), \mathrm{IRS}(p_j), \mathrm{IDRS}(p_j), \mathrm{IIRS}(p_j))$ is called a success node

else if $p_j \in \mathrm{IIRS}(p_i)$, then create a new node $(p_j, \delta(p_j), \mathrm{IRS}(p_j), \mathrm{IDRS}(p_j), \mathrm{IIRS}(p_j))$ in the tree, and an arc, lebeled μ, is directed from the node $(p_i, \delta(p_i), \mathrm{IRS}(p_i), \mathrm{IDRS}(p_i), \mathrm{IIRS}(p_i))$ to the node $(p_j, \delta(p_j), \mathrm{IRS}(p_j), \mathrm{IDRS}(p_j), \mathrm{IIRS}(p_j))$, where $\delta(p_j) = [f_i * \mu, 1 - (t_i * \mu)]$. In this case, the node $(p_j, \delta(p_j), \mathrm{IRS}(p_j), \mathrm{IDRS}(p_j), \mathrm{IIRS}(p_j))$ is called a success node.

Otherwise, for each place $p_k \in \mathrm{IRS}(p_i)$,

if $\mathrm{AP}_{ik} = \phi$ (i.e., p_i does not have any adjacent place) and the goal place $p_j \in \mathrm{RS}(p_k)$ and $\mathrm{CF}_{ik} = \mu$, where $\mu \in [0, 1]$, and p_k does not appear in any node on the path between the root node $(p_s, \delta(p_s), \mathrm{IRS}(p_s), \mathrm{IDRS}(p_s), \mathrm{IIRS}(p_s))$ and the selected node $(p_i, \delta(p_i), \mathrm{IRS}(p_i), \mathrm{IDRS}(p_i), \mathrm{IIRS}(p_i))$, then

if $p_k \in \mathrm{IDRS}(p_i)$, then crate a new node $(p_k, \delta(p_k), \mathrm{IRS}(p_k), \mathrm{IDRS}(p_k), \mathrm{IIRS}(p_k))$ in the tree, and an arc, labeled μ, is directed from the node $(p_i, \delta(p_i), \mathrm{IRS}(p_i), \mathrm{IDRS}(p_i), \mathrm{IIRS}(p_i))$ to the node $(p_k, \delta(p_k), \mathrm{IRS}(p_k, \mathrm{IDRS}(p_k), \mathrm{IIRS}(p_k))$, where $\delta(p_k) = [t_i * \mu, 1 - (f_i * \mu)]$. In this case, the node $(p_k, \delta(p_k), \mathrm{IRS}(p_k), \mathrm{IDRS}(p_k), \mathrm{IIRS}(p_k))$ is called a nonterminal node

else if $p_k \in \mathrm{IIRS}(p_i)$, then create a new node $(p_k, \delta(p_k), \mathrm{IRS}(p_k), \mathrm{IDRS}(p_k), \mathrm{IIRS}(p_k))$ in the tree, and an arc, labeled μ, is directed from the node $(p_i, \delta(p_i), \mathrm{IRS}(p_i), \mathrm{IDRS}(p_i), \mathrm{IIRS}(p_i))$ to the node $(p_k, \delta(p_k), \mathrm{IRS}(p_k), \mathrm{IDRS}(p_k), \mathrm{IIRS}(p_k))$, where $\delta(p_k) = [f_i * \mu, 1 - (t_i * \mu)]$. In this case, the node $(p_k, \delta(p_k), \mathrm{IRS}(p_k), \mathrm{IDRS}(p_k), \mathrm{IIRS}(p_k))$ is called a nonterminal node

else if $\mathrm{AP}_{ik} = \{p_a, p_b, \ldots, p_z\}$ (i.e., $p_a, p_b, \ldots$, and p_z are adjacent places of p_i) and the goal place $p_j \in \mathrm{RS}(p_k)$, then if the truth values of any propositions $d_a, d_b, \ldots$, and d_z are unknown, then request the user to enter the vague truth values of the propositions. Suppose that the vague truth values of the propositions $d_a, d_b, \ldots$, and d_z are $[t_a, 1 - f_a], [t_b, 1 - f_b], \ldots$, and $[t_z, 1 - f_z]$, respectively. Let

$$[t_g, 1 - f_g] = \delta(p_i) \oslash [t_a, 1 - f_a] \oslash [t_b, 1 - f_b] \oslash \ldots \oslash [t_z, 1 - f_z]$$
$$= [t_i, 1 - f_i] \oslash [t_a, 1 - f_a] \oslash [t_b, 1 - f_b] \oslash \ldots \oslash [t_z, 1 - f_z],$$

where "$\oslash$" is the minimum operator between the vague values. That is, $t_g = \mathrm{Min}(t_i, t_a, t_b, \ldots, t_z)$ and $1 - f_g = \mathrm{Min}(1 - f_i, 1 - f_a, 1 - f_b, \ldots, 1 - f_z)$.

If $p_k \in \mathrm{IDRS}(p_i)$ and $\mathrm{CF}_{ik} = \mu$, where $\mu \in [0, 1]$, then create a new node $(p_k, \delta(p_k), \mathrm{IRS}(p_k), \mathrm{IDRS}(p_k), \mathrm{IIRS}(p_k))$ in the tree, and an arc, labeled μ, is directed from the node $(p_i, \delta(p_i), \mathrm{IRS}(p_i), \mathrm{IDRS}(p_i), \mathrm{IIRS}(p_i))$ to the node $(p_k, \delta(p_k), \mathrm{IRS}(p_k), \mathrm{IDRS}(p_k), \mathrm{IIRS}(p_k))$, where $\delta(p_k) = [t_g * \mu, 1 - (f_g * \mu)]$. In this case, the node $(p_k, \delta(p_k), \mathrm{IRS}(p_k), \mathrm{IDRS}(p_k), \mathrm{IIRS}(p_k))$ is called a nonterminal node

else if $p_k \in \mathrm{IIRS}(p_i)$ and $\mathrm{CF}_{ik} = \mu$, where $\mu \in [0, 1]$, then create a new node $(p_k, \delta(p_k), \mathrm{IRS}(p_k), \mathrm{IDRS}(p_k), \mathrm{IIRS}(p_k))$ in the tree, and an arc, labeled μ, is directed from the node $(p_i, \delta(p_i), \mathrm{IRS}(p_i), \mathrm{IDRS}(p_i), \mathrm{IIRS}(p_i))$ to the node $(p_k, \delta(p_k), \mathrm{IRS}(p_k), \mathrm{IDRS}(p_k), \mathrm{IIRS}(p_k))$, where $\delta(p_k) = [f_g * \mu, 1$

- $(t_g * \mu)]$. In this case, the node $(p_k, \delta(p_k), IRS(p_k), IDRS(p_k), IIRS(p_k))$ is called a nonterminal node

else mark the node $(p_i, \delta(p_i), IRS(p_i), IDRS(p_i), IIRS(p_i))$ as a terminal node.

Step 3: If no nonterminal nodes exist, then go to Step 4. Otherwise, go to Step 2.

Step 4: If these are no success nodes, then

(* there does not exist an antecedent-consequence relationship from proposition d_s to proposition d_j *)

stop

else the path from the root node to each success node is called a reasoning path. Let Q be a set of success nodes,

$Q = \{(p_j, [t_1, 1 - f_1], IRS(p_j), IDRS(p_j), IIRS(p_j)),$
$(p_j, [t_2, 1 - f_2], IRS(p_j), IDRS(p_j), IIRS(p_j)), \ldots,$
$(p_j, [t_m, 1 - f_m], IRS(p_j), IDRS(p_j), IIRS(p_j))\},$

where $[t_1, 1 - f_1]$, $[t_2, 1 - f_2]$, ..., and $[t_m, 1 - f_m]$ are vague values in [0, 1]. Let

$[t_j, 1 - f_j] = [t_1, 1 - f_1] \circledvee [t_2, 1 - f_2] \circledvee \ldots \circledvee [t_m, 1 - f_m]$, where "$\circledvee$" is the maximum operator between the vague values. That is

$t_j = \text{Max}(t_1, t_2, \ldots, t_m)$ and
$1 - f_j = \text{Max}(1 - f_1, 1 - f_2, \ldots, 1 - f_m)$.

The degree of truth of proposition d_j is t_j; the degree of false of proposition d_j is f_j, where $t_j + f_j \leq 1$.

4. Examples

In this section, we use an example to illustrate the vague reasoning process of a rule-based system using extended fuzzy Petri nets.

Example 4.1: Let d_1, d_2, d_3, d_4, d_5, d_6, d_7, and d_8 be eight propositions. Assume that the knowledge base of a rule-based system contains the following fuzzy production rules:

R_1: IF d_1 THEN d_2 ELSE d_3 (CF = 0.85)
R_2: IF d_2 THEN d_4 (CF = 0.95)
R_3: IF d_2 THEN d_5 (CF = 0.80)
R_4: IF d_3 THEN d_7 (CF = 0.90)
R_5: IF d_4 or d_5 THEN d_6 (CF = 0.90)
R_6: IF d_6 THEN d_8 (CF = 0.96)

Case 1: Assume that the vague truth value of proposition d_1 given by the user is [0.80, 0.90], and he wants to ask the vague truth value of proposition d_7 might have, then based on [6] the rules and the fact can be modeled by the extended fuzzy Petri net model as shown in Fig. 5. Because of $\beta(p_1) = d_1$ and $\beta(p_7) = d_7$, the places p_1 and p_7 are called the starting place and the goal place, respectively. The immediate direct reachability set, the immediate indirect reachability set, the immediate reachability set, and the reachability set for each place p_i, $p_i \in$ P, in Fig. 5 are shown in Table 4. The adjacent places table for Fig. 5 is shown in Table 5. The certainty factor table for Fig. 5 is shown in Table 6.

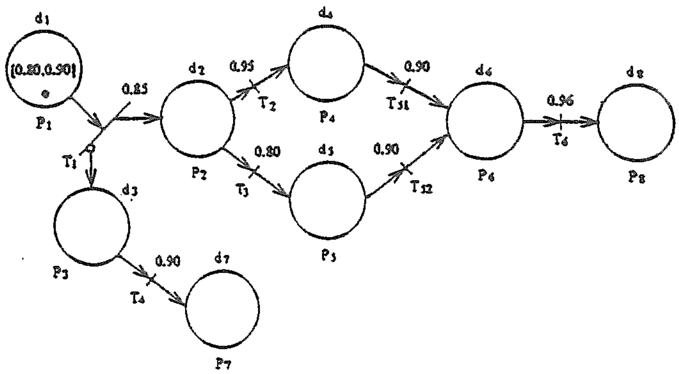

Fig. 5. Marked extended fuzzy Petri net of Example 4.1.

TABLE 4
Immediate Direct Reachability Set, Immediate Indirect Reachability Set, Immediate Reachability Set, and Reachability Set for Each Place p_i in Fig. 5

Place p_i	IDRS(p_i)	IIRS(p_i)	IRS(p_i)	RS(p_i)
p_1	$\{p_2\}$	$\{p_3\}$	$\{p_2, p_3\}$	$\{p_2, p_3, p_4, p_5, p_6, p_7, p_8\}$
p_2	$\{p_4, p_5\}$	ϕ	$\{p_4, p_5\}$	$\{p_4, p_5, p_6, p_8\}$
p_3	$\{p_7\}$	ϕ	$\{p_7\}$	$\{p_7\}$
p_4	$\{p_6\}$	ϕ	$\{p_6\}$	$\{p_6, p_8\}$
p_5	$\{p_6\}$	ϕ	$\{p_6\}$	$\{p_6, p_8\}$
p_6	$\{p_6\}$	ϕ	$\{p_8\}$	$\{p_8\}$
p_7	ϕ	ϕ	ϕ	ϕ
p_8	ϕ	ϕ	ϕ	ϕ

TABLE 5
Adjacent Places Table for Fig. 5

Place p_i	Place p_j	AP_{ij}
p_1	p_2	ϕ
p_1	p_3	ϕ
p_2	p_4	ϕ
p_2	p_5	ϕ
p_3	p_7	ϕ
p_4	p_6	ϕ
p_5	p_6	ϕ
p_6	p_8	ϕ

TABLE 6
Certainty Factor Table for Fig. 6

Place p_i	Place p_j	CF_{ij}
p_1	p_2	0.85
p_1	p_3	0.85
p_2	p_4	0.95
p_2	p_5	0.80
p_3	p_7	0.90
p_4	p_6	0.90
p_5	p_6	0.90
p_6	p_8	0.96

After performing the algorithm, the tree sprouts as shown in Fig. 6. Because there is one success node in the tree, we can obtain the following results:

$$Q = \{(p_7, [0.077, 0.388], \phi, \phi, \phi)\},$$

$$[t_7, 1 - f_7] = [0.077, 0.388].$$

Therefore, the vague truth value of proposition d_7 is [0.077, 0.388]. That is, the degree of truth of proposition d_7 is 0.077; the degree of false of proposition d_7 is 0.612 (i.e., 1 - 0.388 = 0.612).

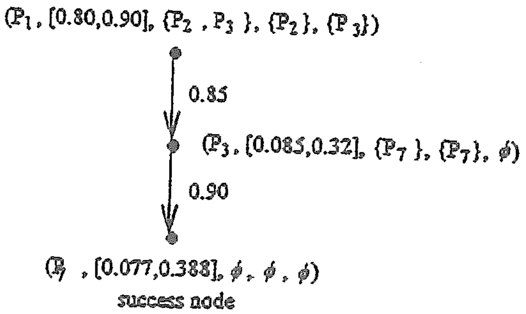

Fig. 6. Sprouting tree of Example 4.1 (Case 1).

Case 2: Assume that the vague truth value of proposition d_1 given by the user is [0.80, 0.90], and he wants to ask the vague truth value proposition d_6 might have. Because $\beta(p_1) = d_1$ and $\beta(p_6) = d_6$, the places p_1 and p_6 are called the starting place and the goal place, respectively. After performing the algorithm, the tree sprouts as shown in Fig. 7. Because there are two success nodes in the tree, we can obtain the following results:

$$Q = \{(p_6, [0.5184, 0.9271], \phi, \phi, \phi), (p_6, [0.4896, 0.9388], \phi, \phi, \phi)\},$$

$$[t_6, 1 - f_6] = [0.5184, 0.9271] \textcircled{$\vee$} [0.4896, 0.9388] = [0.5184, 0.9388].$$

Therefore, the vague truth value of proposition d_6 is [0.5184, 0.9388]. That is, the degree of truth of proposition d_6 is 0.5184; the degree of false of proposition d_6 is 0.0612 (i.e., 1 - 0.9388 = 0.0612).

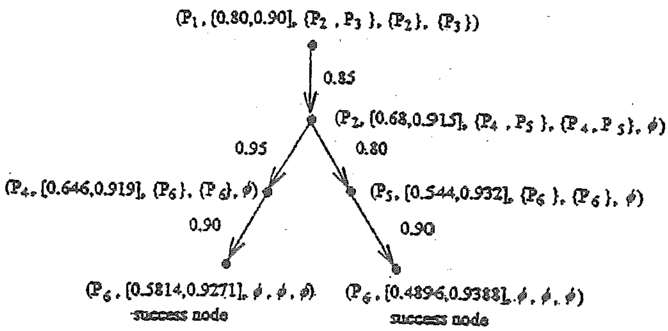

Fig. 7. Sprouting tree of Example 4.1 (Case 2).

5. Conclusions

In this paper, we have extended the work of [5], [6] and [7] to present a vague reasoning algorithm for rule-based systems based on the extended fuzzy Petri nets, where fuzzy IF-THEN rules and fuzzy IF-THEN-ELSE rules are used for knowledge representation, and the truth values of the propositions appearing in the rules are represented by vague values in [0, 1] rather than real values between zero and one. The proposed vague reasoning algorithm is more flexible than the one presented in [7] due to the fact that the proposed algorithm allows fuzzy IF-THEN-ELSE rules to be used for knowledge representation and allows the truth values of the propositions appearing in the rules to be represented by vague values in [0, 1] rather than real values between zero and one. The upper bound of the time complexity of the proposed vague reasoning algorithm is $O(nm)$, where n is the number of places and m is the number of transitions in an extended fuzzy Petri nets. Its execution time is propotional to the number of nodes in a sprouting tree generated by the proposed algorithm. This vague reasoning capability allows the computers to perform reasoning in a more flexible and more intelligent manner.

References

[1] A. J. Bugarin and S. Barro, "Fuzzy reasoning supported by Petri nets," IEEE Transactions on Fuzzy Systems, vol. 2, no. 2, pp. 135-150, May 1994.

[2] T. Cao and A. C. Sanderson, "Task sequence planning using fuzzy Petri nets," IEEE Transactions on Systems, Man, and Cybernetics, vol. 25, no. 5, pp. 755-768, May 1995.

[3] C. L. Chang, Introduction to Artificial Intelligence Techniques, JMA Press, 1985.

[4] S. M. Chen, "A new approach to handling fuzzy decisionmaking problems," IEEE Transactions on Systems, Man, Cybernetics, vol. 18, no. 6, pp. 1012-1016. November/December 1988,.

[5] S. M Chen, "Vague reasoning techniques for rule-based systems," Proceedings of 1995 National Computer Symposium, Chungli, Taoyuan, Taiwan, R. O. C., vol. 2, pp. 589-595. December 1995.

[6] S. M. Chen, "Knowledge representation using extended fuzzy Petri nets," Proceedings of 1996 International Conference on Artificial Intelligence (ICS'96), Kaohsiung, Taiwan, R. O. C., December 1996.

[7] S. M. Chen, J. S. Ke, and J. F. Chang, "Knowledge representation using fuzzy Petri nets," IEEE Trans. Knowledge and Data Engineering, vol. 2, no. 3, pp. 311-319, September 1990.

[8] M. L. Garg, S. I. Ahson, and P. V. Gupta, "A fuzzy Petri net for knowledge representation and reasoning," Information Processing Letters, vol. 39, pp. 165-171, 1991.

[9] W. L Gau, and D. J. Buehrer, "Vague sets," IEEE Trans. Syst., Man, Cybern., vol. 23, no. 2, pp. 610-614, March/April 1993.

[10] A. Giordana, and L. Saitta, "Modeling production rules by means of predicate transition networks," Inform. Sci., vol. 35, no. 1, pp. 1-41, 1985.

[11] A. Kandel, Fuzzy Mathematical Techniques with Applications, Addison-Wesley, 1986.

[12] A. Kaufmann, and M. M. Gupta, Fuzzy Mathematical Models in Engineering and Management Science, North-Holland, 1988.

[13] A. Konar and A. K. Mandal, "Uncertainty management in expert systems using fuzzy Petri nets," IEEE Transactions on Knowledge and Data Engineering, vol. 8, no. 1, pp. 96-105, February 1996.

[14] C. G. Looney, "Fuzzy Petri nets for rule-based decisionmaking," IEEE Trans. Syst., Man, Cybern., vol. 18, no. 1, pp. 178-183, January/February 1988.

[15] C. G. Looney, "Fuzzy Petri nets and applications," in Fuzzy Reasoning in Information, Decision, and Control Systems, edited by S. G. Tzafestas and A. N. Venetsanopoulos, Kluwer Academic Publishers, pp. 511-527, 1994.

[16] J. L.Peterson, Petri Nets, Theory and The Modeling of Systems, Prentice-Hall, 1981.

[17] W. Pedrycz and F. Gomide, "A generalized fuzzy Petri net model," IEEE Transactions on Fuzzy Systems, vol. 2, no. 4, pp. 295-301, November 1994.

[18] S. K. Yu, "Knowledge representation and reasoning using Pr/T net-systems," Fuzzy Sets and Systems, vol. 75, no. 1, pp. 33-45, October 1995.

[19] L. A. Zadeh, "Fuzzy sets," Infor. Contr., vol. 8, pp. 338-353, 1965.

[20] L. A. Zadeh, "Fuzzy logic," IEEE Computer Magazine, vol. 21, no. 4, pp. 83-93, April 1988.

[21] H.-J. Zimmermann, Fuzzy Set Theory and Its Applications, Kluwer-Nijhoff, 1991.