Intuitionistic Fuzzy Petri Nets for Knowledge Representation and Reasoning

Meng Fei-xiang[1], Lei Ying-jie[1], Zhang Bo[1], Shen Xiao-yong[1], Zhao Jing-yu[2]
[1]Air and Missile Defense College
Air Force Engineering University, Xi'an, 710051, China
[2]School of Electrical, Computer and Telecommunications Engineering
University of Wollongong
Wollongong NSW 2522, Australia
ttimo@163.com

Journal of Digital Information Management

***ABSTRACT**: Fuzzy Petri nets (FPNs) are an ideal modeling tool for knowledge-based systems, which are based on fuzzy production rules. FPNs are widely used in knowledge representation and reasoning, assessment, fault diagnosis, exception handling, and other fields, but they have the defects of single membership degree. To solve this problem, intuitionistic fuzzy Petri nets (IFPNs) were presented for knowledge representation and reasoning. First, the IFPN model was constructed for knowledge representation by combining intuitionistic fuzzy sets theory with Petri nets theory. Second, an algorithm based on IFPN was proposed, and the matrix operation was introduced into the reasoning process to make full use of the parallel computing capability of Petri nets. Finally, an example was illustrated to prove the feasibility and advantages of the proposed IFPN model and reasoning algorithm. Moreover, the reasoning result was analyzed and discussed. Compared with FPN, IFPN can describe three states, namely, the support state, the opposite state, and the neutral state. Thus, IFPN can overcome the single membership degree of FPN and describe the reasoning result more comprehensively and precisely than FPN. Moreover, IFPN is an effective extension and development of FPN and will become a promising method for knowledge representation and reasoning.*

Subject Categories and Descriptors
I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods- Representations(procedural and rule-based)

General Terms: Algorithm, Performance

Keywords: Intuitionistic fuzzy set, Intuitionistic fuzzy Petri net, Knowledge representation, Intuitionistic fuzzy reasoning

Received: 1 November 2015, Revised: 27 December 2015, Accepted: 30 December 2015

1. Introduction

Artificial intelligence (AI) is a frontier discipline. AI is widely used because it can simulate humans to deal with problems and has a certain learning ability. Knowledge is the basis of AI, so knowledge representation has become one of the most important research topics in the field of AI [1]. In recent years, many knowledge representation methods have been presented, such as first-order predicate logic, production rules, fuzzy production rules (FPRs), frames, semantic networks, scripts, and Petri nets. In this paper, we propose an intuitionistic fuzzy Petri net (IFPN) model to represent the intuitionistic fuzzy production rules (IFPRs) of a knowledge-based system. The IFPN allows structured representation of knowledge, and the reasoning process can make full use of the parallel operation ability of Petri nets. The IFPN can fully describe the reasoning results because it increases a nonmembership function.

The remainder of this paper is organized as follows. Section 2 briefly reviews the relevant works of fuzzy Petri nets (FPNs). Section 3 proposes a knowledge representation method using IFPN and a reasoning algorithm based on IFPN. Section 4 illustrates an example to prove the feasibility and validity of the proposed method, as well as analyzes the result. Conclusions are summarized in Section 5.

2. Related Works

FPNs are an ideal modeling tool for knowledge-based systems, which are based on FPRs. Since FPN was proposed, knowledge representation and reasoning methods based on FPN have been extensively investigated. Chen proposed a method for knowledge representation and reasoning based on FPN [2] and then proposed the weighted fuzzy Petri net by introducing the weight into FPN [3]. Gao et al. [4] studied the representation method of negative propositions in fuzzy reasoning Petri nets (FRPNs), as well as proposed a reasoning algorithm for FRPNs based on matrix operation. Wang et al. [5] pointed out the inconsistent representation method of negative propositions in [4] and subsequently proposed a consistent fuzzy Petri net, which was more suitable to represent fuzzy logic programs with negations. Jia et al. [6] proposed a formalized reasoning algorithm based on matrix operations making full use of the parallel commutating capability of Petri nets. However, the truth degree of consequent propositions calculated by this method may be larger than 1.

Considering the increasing complexity in knowledge representation, the traditional FPN model cannot meet the requirements of knowledge representation very well. A number of scholars have proposed many expanded FPN models. Li et al. [7] proposed adaptive fuzzy Petri nets with learning ability. Liu et al. [1][8] proposed a dynamic adaptive fuzzy Petri net model. The model has a dynamically adaptive ability, so it can represent the knowledge-based expert system more correctly. Shen [9] proposed a high-level fuzzy Petri net model, which can solve negative problems because it represents IF-THEN and IF-THEN-ELSE rules simultaneously. Ribaric et al. [10] proposed a high-level Petri net model for fuzzy spatio-temporal knowledge representation and reasoning. Amin [11] introduced a new class of fuzzy Petri nets that consider the weight changes of the arc in the fuzzy reasoning process. Pedrycz [12] and Liu [13] proposed a fuzzy timed Petri net model by introducing the time factor into FPN. Wang [14] proposed a fuzzy colored time Petri net.

In recent years, FPN has been widely used in the areas of knowledge representation and reasoning [1][2][3][4][7][8], assessment [15], fault diagnosis [16][17], control system [18][19], and exception handling [20]. However, research on FPN is almost limited to combine Zadeh fuzzy sets (ZFS) with Petri nets theory. In ZFS, only a single scale (i.e., membership degree or membership function) is used to define a fuzzy set, so ZFS can only describe the fuzzy concept of “either/or” and cannot represent the neutral state. When FPN inherits the advantages of ZFS, it also inherits the single membership degree limitation of FPN. However, intuitionistic fuzzy set (IFS) [21] adds a nonmembership function, so it can describe the neutral state and the fuzzy features of the real world more delicately and fully. IFS is the most effective extension and development of ZFS.

Therefore, we combine IFS theory and Petri nets theory to construct an IFPN model for knowledge representation and reasoning to solve the single membership degree limitation in FPN and explore a novel method for knowledge representation and reasoning.

3. Methodology

3.1 IFS and Basic Operation

The definition of IFS and the basic operation are presented by Atanassov [21] as follows:

Definition 1 (IFS): Let X be the universe of discourse, $X = \{x_1, x_2, \cdots, x_n\}$. An IFS A of X is defined as

$$A = \{< x, \mu_A(x), \gamma_A(x) > | x \in X\} \tag{1}$$

where $\mu_A(x): X \to [0,1]$ is the membership function of A, $\gamma_A(x): X \to [0,1]$ is the nonmembership function of A, and for every $x \in X$ in A, $0 \le \mu_A(x) + \gamma_A(x) \le 1$ holds.

When X is a continuous space,

$$A = \int_X < \mu_A(x), \gamma_A(x) > /x, x \in X \tag{2}$$

When X is a discrete space,

$$A = \sum_{i=1}^{n} < \mu_A(x_i), \gamma_A(x_i) > /x_i, x_i \in X \tag{3}$$

For every subset A of X, $\pi_A(x) = 1 - \mu_A(x) - \gamma_A(x)$ is called an intuitionistic index of x in A, and it is a measurement of hesitancy degree about A from x. Obviously, for every $x \in X$, $0 \le \pi_A(x) \le 1$.

Definition 2 (basic operation of IFS [21]): Let A and B be the subsets of the universe of discourse X, then

(1) $A \cap B = \{< x, \mu_A(x) \wedge \mu_B(x), \gamma_A(x) \vee \gamma_B(x) > | \forall x \in X\}$

(2) $A \cup B = \{\langle x, \mu_A(x) \vee \mu_B(x), \gamma_A(x) \wedge \gamma_B(x)\rangle | \forall x \in X\}$

(3) $\overline{A} = A^c = \{\langle x, \gamma_A(x), \mu_A(x)\rangle | x \in X\}$

(4) $A \subseteq B \Leftrightarrow \forall x \in X, [\mu_A(x) \leqslant \mu_B(x) \wedge \gamma_A(x) \geqslant \gamma_B(x)]$

(5) $A \subset B \Leftrightarrow \forall x \in X, [\mu_A(x) < \mu_B(x) \wedge \gamma_A(x) > \gamma_B(x)]$

(6) $A = B \Leftrightarrow \forall x \in X, [\mu_A(x) = \mu_B(x) \wedge \gamma_A(x) = \gamma_B(x)]$

3.2 Knowledge representation using IFPN

In this section, the formal definition of IFPN is given, and the IFPN is used for knowledge representation.

3.2.1 Definition of IFPN

The IFPN model is defined as follows:

Definition 3 $\text{IFPN}=(P, T, D, I, O, \delta, \theta, Th, CF)$, where

(1) $P = \{p_1, p_2, \cdots, p_n\}$ is a finite set of places.

(2) $T = \{t_1, t_2, \cdots, p_m\}$ is a finite set of transitions.

(3) $D = \{d_1, d_2, \cdots, d_n\}$ is a finite set of propositions, $|P| = |D|, P \cap T \cap D = \varnothing$.

(4) $I: P \times T \to \{0,1\}$ is an $n \times m$ input matrix. If there is an input directed arc from p_i to t_j, then $I(p_i, t_j) = 1$, otherwise $I(p_i, t_j) = 0$, where $i = 1,2,\cdots,n,\ j = 1,2,\cdots,m$.

(5) $O: T \times P \to \{0,1\}$ is an $m \times n$ output matrix. If there is an output directed arc from t_j to p_i, then $O(t_j, p_i) = 1$, otherwise $O(t_j, p_i) = 0$, where $i = 1,2,\cdots,n,\ j = 1,2,\cdots,m$.

(6) $\delta: P \to D$ represents the relation of places and propositions.

(7) $\theta = (\theta_1, \theta_2, \cdots, \theta_n)^T$ is a vector. $\theta_i = (\mu_i, \gamma_i)$ is an intuitionistic fuzzy number and indicates the truth degree of p_i, where $\mu_i \in [0,1]$, $\theta_i \in [0,1]$ and $\mu_i + \theta_i \in [0,1]$. The initial truth degree vector is denoted by $\theta^0 = (\theta_1^0, \theta_2^0, \cdots, \theta_n^0)^T$, where $\theta_2^0 =< \mu_i^0, \gamma_i^0 >$.

(8) $Th = (\lambda_1, \lambda_2, \cdots, \lambda_m)^T$ is a vector. $\lambda_i = (\alpha_i, \beta_i)$ is an intuitionistic fuzzy number and represents the threshold value of transitions, where $\alpha_i \in (0,1]$, $\beta_i \in (0,1]$ and $\alpha_i + \beta_i \in (0,1]$.

(9) $CF = diag(CF_1, CF_2, \cdots, CF_m)$, $CF_j =< C\mu_j, C\gamma_j >$ is the certainty factor of rule R_j, where $C\mu_j \in (0,1]$, $C\gamma_j \in (0,1]$ and $C\mu_j + C\gamma_j \in (0,1]$.

3.2.2 Knowledge representation using IFPN

Production rules are very simple and convenient for reasoning, so they are usually used for knowledge representation. IFPRs are combined by production rules and IFS, so they are used for both knowledge representation and reasoning.

Let $R = \{R_1, R_2, \cdots, R_n\}$ be a set of IFPR, such that the general formulation of the ith IFPR has the form:

$$R_i: \text{IF } d_j \text{ THEN } d_k \ (CF_i, \lambda_i)$$

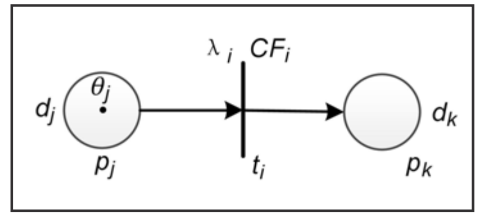
where d_j and d_k are intuitionistic fuzzy proposition and represent the antecedent and consequence of rule R_i, respectively; their truth degrees are θ_j and θ_k, respectively. CF_i and λ_i represent the certainty factor and threshold value of rule R_i, respectively. θ_j , θ_k, CF_i and λ_i are intuitionistic fuzzy numbers. The corresponding relation between the IFPR set and IFPN model is shown in Table 1.

IFPR can be categorized into four types. Their IFPN model and the conditions of transition firing and the token value's transitional rules after the transition is fired are as follows.

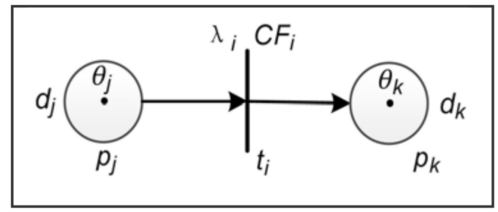
(1) Type 1: A simple IFPR and its IFPN model

$$R_i: \text{IF } d_j \text{ THEN } d_k \ (CF_i, \lambda_i)$$

Let $\theta_j =< \mu_j, \gamma_j >$, $\theta_k =< \mu_k, \gamma_k >$, $\lambda_i =< \alpha_i, \beta_i >$, $CF_i =< C\mu_i, C\gamma_i >$, then



(a) before the transition is fired



(b) after the transition is fired

Figure 1. IFPN model of type 1

IFPR set	IFPN model
rule R_j	transition t_j
the antecedent of R_j	the input place of t_j
the consequence of R_j	the output place of t_j
proposition d_k	place p_k
the truth degree of d_k	the token value of p_k
the threshold value of R_j	threshold value of t_j
the certainty factor of R_j	the certainty factor of t_j
R_j is applied	t_j is fired

Table 1. Corresponding relation between IFPR set and IFPN model

1) conditions of the transition being fired

If and only if $\begin{cases} \mu_j \ge \alpha \\ \gamma_j \le \beta \end{cases}$, t_i can be fired;

2) the token value's transitional rules after the transition is fired

$$\begin{cases} \mu_k = \mu_j \times C\mu_i \\ \gamma_k = \gamma_j + C\gamma_i - \gamma_j \times C\gamma_i \end{cases} \tag{4}$$

(2) Type 2: A composite intuitionistic fuzzy conjunctive rule in the antecedent and its IFPN model

$$R_i: \text{IF } d_{j1} \text{ AND } d_{j2} \text{ AND} \cdots \text{AND } d_{jn} \text{ THEN } d_k \ (CF_i, \lambda_i)$$

Let $\theta_{jm} =< \mu_{jm}, \gamma_{jm} >$ $(m = 1,2,\cdots,n)$, $\theta_k =< \mu_k, \gamma_k >$, $\lambda_i =< \alpha_i, \beta_i >$, $CF_i =< C\mu_i, C\gamma_i >$, then

1) conditions of the transition being fired

If and only if $\begin{cases} \min(\mu_{j1}, \mu_{j2}, \cdots, \mu_{jn}) \ge \alpha_i \\ \max(\gamma_{j1}, \gamma_{j2}, \cdots, \gamma_{jn}) \le \beta_i \end{cases}$, t_i can be fired;

2) the token value's transitional rules after the transition is fired

$$\begin{cases}\mu_k = \min(\mu_{j1}, \mu_{j2}, \cdots, \mu_{jn}) \times C\mu_i \\ \gamma_k = \max(\gamma_{j1}, \gamma_{j2}, \cdots, \gamma_{jn}) + C\gamma_i - \max(\gamma_{j1}, \gamma_{j2}, \cdots, \gamma_{jn}) \times C\gamma_i \end{cases} \quad (5)$$

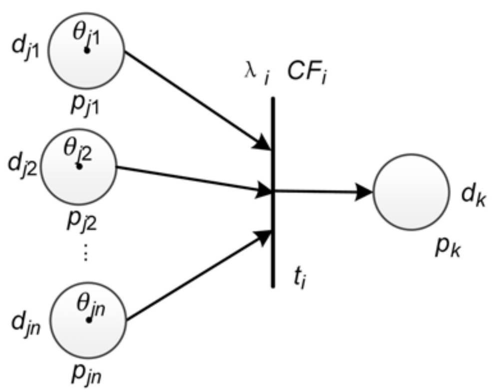
(3) Type 3: A composite intuitionistic fuzzy disjunctive rule in the antecedent and its IFPN model

$$R_i: \text{IF } d_{j1} \text{ OR } d_{j2} \text{ OR} \cdots \text{OR } d_{jn} \text{ THEN } d_k \ (CF_i, \lambda_i)$$

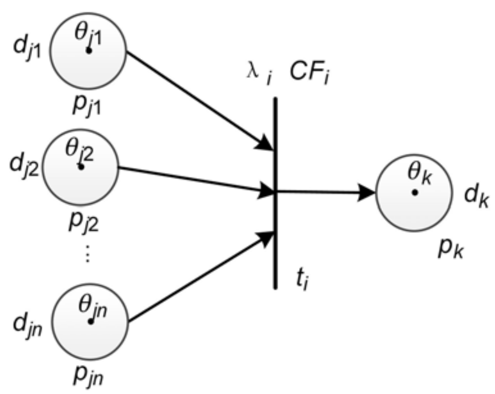
R_i can be converted into n rules as follows:

$$R_{i1}: \text{IF } d_{j1} \text{ THEN } d_k \ (CF_i, \lambda_i)$$
$$R_{i2}: \text{IF } d_{j2} \text{ THEN } d_k \ (CF_i, \lambda_i)$$
$$\cdots\cdots$$
$$R_{in}: \text{IF } d_{jn} \text{ THEN } d_k \ (CF_i, \lambda_i)$$

Let $\theta_{jm} = <\mu_{jm}, \gamma_{jm}> (m = 1, 2, \cdots, n)$, $\theta_k = <\mu_k, \gamma_k>$, $\lambda_i = <\alpha_i, \beta_i>$, $CF_i = <C\mu_i, C\gamma_i>$, then



(a) before the transition is fired

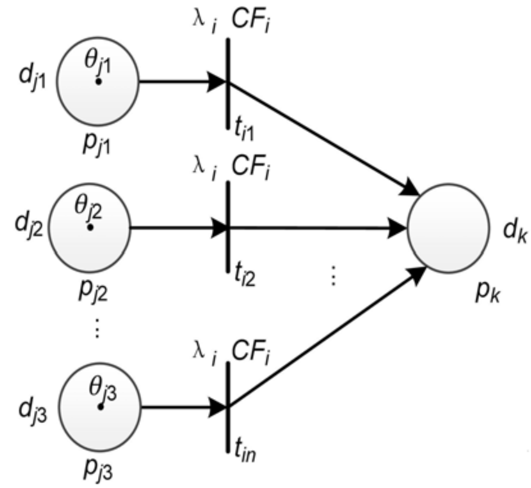


(b) after the transition is fired

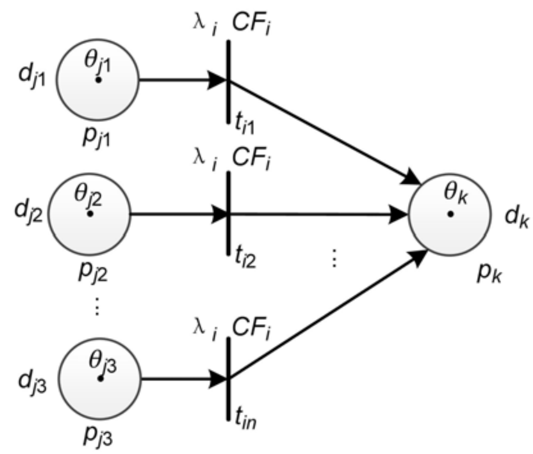
Figure 2. IFPN model of type 2

1) conditions of the transition being fired

Places $p_{j1}, p_{j2}, \cdots, p_{jm} (1 \le m \le n)$ meet $\begin{cases}\mu_{jm} \ge \alpha_i \\ \gamma_{jm} \le \beta_i\end{cases}$ at the same time, t_i can be fired;



(a) before the transition is fired



(b) after the transition is fired

Figure 3. IFPN model of type 3

2) the token value's transitional rules after the transition is fired

$$\begin{cases}\mu_k = \max(\mu_{j1} \times C\mu_i, \mu_{j2} \times C\mu_i, \cdots, \mu_{jm} \times C\mu_i) \\ \gamma_k = \min(\gamma_{j1} + C\gamma_i - \gamma_{j1} \times C\gamma_i, \gamma_{j2} + C\gamma_i - \gamma_{j2} \times C\gamma_i, \cdots, \gamma_{jm} + C\gamma_i - \gamma_{jm} \times C\gamma_i)\end{cases} \quad (6)$$

(4) Type 4: A composite intuitionistic fuzzy conjunctive rule in the consequence and its IFPN model

$$R_i: \text{IF } d_j \text{ THEN } d_{k1} \text{ AND } d_{k2} \text{ AND} \cdots \text{AND } d_{kn} \ (CF_i, \lambda_i)$$

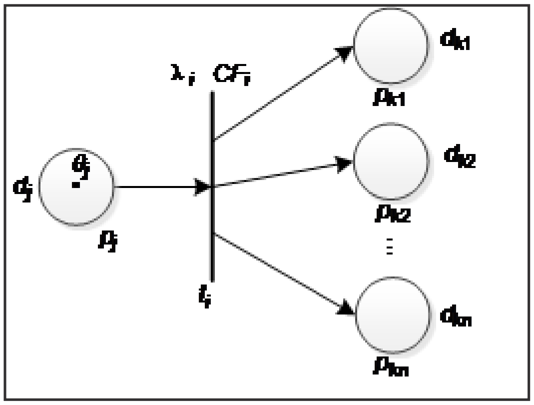
Let $\theta_j = <\mu_j, \gamma_j>$, $\theta_{ki} = <\mu_{ki}, \gamma_{ki}> (i = 1, 2, \cdots, n)$, $\lambda_i = <\alpha_i, \beta_i>$, $CF_i = <C\mu_i, C\gamma_i>$, then

1) conditions of the transition being fired

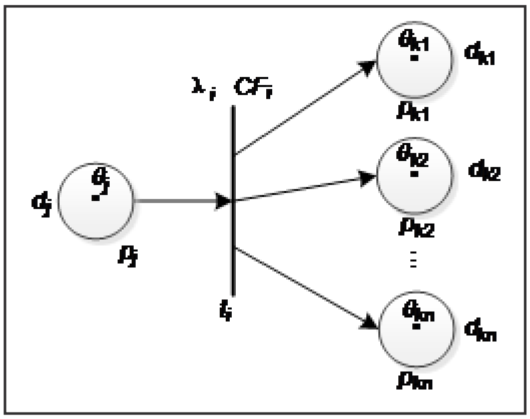
If and only if $\begin{cases}\mu_j \ge \alpha_i \\ \gamma_j \le \beta_i\end{cases}$, t_i can be fired;

2) the token value's transitional rules after the transition is fired

$$\begin{cases}\mu_{ki} = \mu_j \times C\mu_i \\ \gamma_{ki} = \gamma_j + C\gamma_i - \gamma_j \times C\gamma_i\end{cases} \quad (7)$$



(a) before the transition is fired



(b) after the transition is fired

Figure 4. IFPN model of type 4

3.3 Reasoning algorithm based on IFPN

3.3.1 Definition of Operators

To represent the reasoning algorithm formally, some operators are defined.

(1) multiplication operator $\otimes$:

$C = A \otimes B$, where, $A = (a_{ij})_{m\times n}$, $B = (b_{ij})_{m\times n} = (<\mu b_{ij}, \gamma b_{ij}>)_{m\times n}$,

$C = (c_i)_{m\times 1} = (<c\mu_i, c\gamma_i>)_{m\times 1}$,

$$c\mu_i = \max\left\{x_i \mid x_i = \begin{cases} b\mu_j, a_{ij} = 1 \\ 0, a_{ij} = 0 \end{cases}\right\}, c\gamma_i = \min\left\{y_i \mid y_i = \begin{cases} b\gamma_j, a_{ij} = 1 \\ 1, a_{ij} = 0 \end{cases}\right\}$$

(2) addition operator $\oplus$:

$C = A \oplus B$, where $C = (c_{ij})_{m\times n} = (<\mu c_{ij}, \gamma c_{ij}>)_{m\times n}$,

$A = (a_{ij})_{m\times n} = (<\mu a_{ij}, \gamma a_{ij}>)_{m\times n}$,

$B = (b_{ij})_{m\times n} = (<\mu b_{ij}, \gamma b_{ij}>)_{m\times n}$,

$c_{ij} = \max(a_{ij}, b_{ij}) = (\max <\mu a_{ij}, \mu b_{ij}>, \min <\gamma a_{ij}, \gamma b_{ij}>)$

(3) compare operator $\oslash$:

$C = A \oslash B$, where

$$C = (c_1, c_2, \cdots, c_m)^T = (<\mu c_1, \gamma c_1>, <\mu c_2, \gamma c_2>, \cdots, <\mu c_m, \gamma c_m>)^T ,$$

$$A = (a_1, a_2, \cdots, a_m)^T = (<\mu a_1, \gamma a_1>, <\mu a_2, \gamma a_2>, \cdots, <\mu a_m, \gamma a_m>)^T ,$$

$$B = (b_1, b_2, \cdots, b_m)^T = (<\mu b_1, \gamma b_1>, <\mu b_2, \gamma b_2>, \cdots, <\mu b_m, \gamma b_m>)^T ,$$

$$c_i = <\mu c_i, \gamma c_i> = \begin{cases} <\mu a_i, \gamma a_i>, \mu c_i \geq \mu a_i \text{ and } \gamma c_i \leq \gamma a_i \\ <0,1>, other \end{cases}$$

(4) direct multiplication operator Θ:

$C = A \Theta B$, where $C = (c_{ij})_{m\times n} = (<\mu c_{ij}, \gamma c_{ij}>)_{m\times n}$,

$A = (a_{ij})_{m\times l} = (<\mu a_{ij}, \gamma a_{ij}>)_{m\times l}$,

$B = (b_{ij})_{l\times n} = (<\mu b_{ij}, \gamma b_{ij}>)_{l\times n}$,

$c_{ij} = (c_{ij})_{m\times n} = <\mu a_{ij} \times \mu b_{ij}, \gamma a_{ij} + \gamma b_{ij} - \gamma a_{ij} \times \gamma b_{ij}>$

(5) negative operator neg

Let $\theta = (\theta_1, \theta_2, \cdots, \theta_n)^T = (<\mu_1, \gamma_1>, <\mu_2, \gamma_2>, \cdots, <\mu_n, \gamma_n>)^T$, then

$$\begin{aligned} neg\theta &= neg(\theta_1, \theta_2, \cdots, \theta_n)^T \\ &= neg(<\mu_1, \gamma_1>, <\mu_2, \gamma_2>, \cdots, <\mu_n, \gamma_n>)^T \\ &= (<\gamma_1, \mu_1>, <\gamma_2, \mu_2>, \cdots, <\gamma_n, \mu_n>)^T = \overline{\theta} \end{aligned}$$

3.3.2 Reasoning algorithm based on IFPN

An acyclic net is a net in which no loop or circuit exists. No circularity exists in most practically used knowledge bases [4]. Therefore, in this paper, the IFPN of IFPR is assumed to be an acyclic net.

Definition 4 (pre-set, post-set)

Let $\text{IFPN} = (P, T, F; I, O, \theta, Th, CF)$ be an IFPN, then

${}^{\bullet}p = \{t \mid (t, p) \in F\}$ is called pre-set of the place p or input transition set;

$p^{\bullet} = \{t \mid (p, t) \in F\}$ is called post-set of the place p or output transition set;

${}^{\bullet}t = \{p \mid (p, t) \in F\}$ is called pre-set of the transition t or input place set;

$t^{\bullet} = \{p \mid (t, p) \in F\}$ is called post-set of the transition t or output place set.

Definition 5 (immediate reachability set, reachability set [2])

In the IFPN model, let t_i be a transition and p_i, p_j, p_k be places. If $p_i \in {}^{\bullet}t_i$ and $p_j \in t_i^{\bullet}$, then p_j is called immediately reachable from p_i and is denoted by $p_i \Rightarrow p_j$;if $p_i \Rightarrow p_j, p_j \Rightarrow p_{j+1}, \cdots, p_{j+k-1} \Rightarrow p_{j+k}$,then $p_{j+1}, p_{j+2}, \cdots, p_{j+k}$ is called reachable from p_i and is denoted by $p_i \rightarrow p_{j+1}, p_i \rightarrow p_{j+2}, \cdots, p_i \rightarrow p_{j+k}$.The set of places immediately reachable from p_i is called immediate reachability set of p_i and is denoted by $IRS(p_i)$. The set of places reachable from p_i is called reachability set of p_i and is denoted by $RS(p_i)$.

In case there are n propositions and m rules in IFPR set S , so there are n places and m transitions in the corresponding IFPN model, the reasoning algorithm based on IFPN is as follows:

Algorithm 1 Reasoning algorithm based on IFPN

Input: input matrix I output matrix O, threshold value of transitions Th, certainty factor of rules CF, initial truth degree of propositions θ^0.

Output: token value of places (i.e., the real value of proposition) and the iteration times k.

Preprocess (to determine whether circuits exist in the IFPN model): In the IFPN model, if $\exists p_i \in RS(p_i)$, then circuits exist in the model. Then, the reasoning algorithm cannot be applied in the model, and quit.

Step 1 Initialize all the inputs, and let $k=1$,

$\theta^{k-1}=\theta^0=(\theta_1^0,\theta_2^0,\cdots,\theta_n^0)^T=(<\mu_1^0,\gamma_1^0>,<\mu_2^0,\gamma_2^0>,\cdots,<\mu_n^0,\gamma_n^0>)^T$.

The truth degree of each unknown proposition is indicated by $<0,1>$, and the initial equivalent input is $\rho_{k-1}=\rho_0=(<0,1>,<0,1>,\cdots,<0,1>)^T$.

Step 2 Calculate the equivalent input of each transition, i.e., the token value of all the input places of each transition should be equivalent to the token value of a single input place, and the final result is

$\rho_k=(\rho\theta_1^k,\rho\theta_2^k,\cdots,\rho\theta_m^k)^T=(<\rho\mu_1^k,\rho\gamma_1^k>,<\rho\mu_2^k,\rho\gamma_2^k>,\cdots,<\rho\mu_m^k,\rho\gamma_m^k>)^T$ where

$$\rho\mu_j^k=\min\left\{x_i \mid x_i=\begin{cases}\mu_i^{k-1}, I(p_i,t_j)=1\\ 1, I(p_i,t_j)=0\end{cases}\right\},\ \rho\gamma_j^k=\max\left\{y_i \mid y_i=\begin{cases}\gamma_i^{k-1}, I(p_i,t_j)=1\\ 0, I(p_i,t_j)=0\end{cases}\right\}.$$

i.e., $\rho_k=\overline{(I^T\otimes\overline{\theta_{k-1}})}$ (8)

Step 3 Compare the equivalent input token value of each transition with the threshold value of each rule. Keep the equivalent input token value, which can fire the transition:

$$\rho_k'=\rho_k\oslash Th \qquad (9)$$

Step 4 Calculate the token values of output places after the transitions are fired

1) $S_k=CF\Theta\rho_k'$, where

$S_k=(s_1^k,s_2^k,\cdots,s_m^k)^T=(<s\mu_1^k,s\gamma_1^k>,<s\mu_2^k,s\gamma_2^k>,\cdots,<s\mu_n^k,s\gamma_m^k>)^T$

$$\begin{cases}s\mu_j^k=C\mu_j^k\times\rho\mu_j^k\\ s\gamma_j^k=C\gamma_j^k+\rho\gamma_j^k-C\gamma_j^k\times\rho\gamma_j^k\end{cases}$$

2). $Y_k=(y\theta_1^k,y\theta_2^k,\cdots,y\theta_n^k)^T=(<y\mu_1^k,y\gamma_1^k>,<y\mu_2^k,y\gamma_2^k>,\cdots,<y\mu_n^k,y\gamma_n^k>)^T$,

where

$$y\mu_i^k=\max\left\{x_i \mid x_i=\begin{cases}s\mu_j^k, O(p_i,t_j)=1\\ 0, O(p_i,t_j)=0\end{cases}\right\},\ y\gamma_i^k=\min\left\{y_i \mid y_i=\begin{cases}s\gamma_j^k, O(p_i,t_j)=1\\ 1, O(p_i,t_j)=0\end{cases}\right\},$$

i.e., $Y_k=(O\otimes S_k)=O\otimes(CF\Theta\rho_k')$ (10)

Step 5 Calculate the token values of all places (i.e., the final truth degrees of all propositions):

$$\begin{aligned}\theta^k&=\theta^{k-1}\oplus Y_k=(\theta_1^k,\theta_2^k,\cdots,\theta_n^k)^T\\ &=(<\mu_1^k,\gamma_1^k>,<\mu_2^k,\gamma_2^k>,\cdots,<\mu_n^k,\gamma_n^k>)^T\end{aligned} \qquad (11)$$

Step 6 Assess whether the reasoning process is finished. If $\theta^k=\theta^{k-1}$, the reasoning process is finished, and the final value of all proposition θ^k is used as output. Otherwise, make $k=k+1$ and proceed to Step 2.

3.3.3 Algorithm analysis

Definition 6 (Source Place, Sink Place)
If ${}^{\bullet}p=\{t|(t,p)\in F\}=\varnothing$, then place p called a source place. If $p^{\bullet}=\{t|(p,t)\in F\}=\varnothing$, then place p called a sink place.

Definition 6 (Route[7])
In an IFPN model, there are n places and m transitions, given a place p, a transition string $t_1,t_2,\cdots,t_n$ is called a route of p if p can obtain token values from a group of source places by firing $t_1,t_2,\cdots,t_n$ in sequence.

Theorem 1 $\theta^k=\theta^{k-1}$ is the necessary and sufficient condition of the reasoning process being finished.

This theorem is established, so the proving process is skipped.

Theorem 2 This reasoning algorithm can be finished after k times rotation, where $1\le k\le h+1$, and h indicates the transition number of the longest route in the IFPN model.

Proof: (1) First, we prove that the algorithm can be finished after k times rotation.

Suppose $\theta^k=\theta^{k-1}$ when the reasoning process is finished at the h^{th} rotation. According to Theorem 1, the reasoning process is completed.

(2) Second, we prove $1\le k\le h+1$.

(i) Initially, we prove $k=h+1$

Suppose h indicates the transition number of the longest route in the IFPN model, so we only need to prove when $k=h+1$, $\theta^{h+1}=\theta^h$ after the reasoning process has been finished.

$\theta^h=(\theta_1^h,\theta_2^h,\cdots,\theta_n^h)^T=(<\mu_1^h,\gamma_1^h>,<\mu_2^h,\gamma_2^h>,\cdots,<\mu_n^h,\gamma_n^h>)^T$
and
$\theta^{h+1}=(\theta_1^{h+1},\theta_2^{h+1},\cdots,\theta_n^{h+1})^T=(<\mu_1^{h+1},\gamma_1^{h+1}>,<\mu_2^{h+1},\gamma_2^{h+1}>,\cdots,<\mu_n^{h+1},\gamma_n^{h+1}>)^T$
is known.

Suppose p_i is the Sink place of the longest route, the corresponding transition is t_i. When the reasoning process proceeds to the hth step and h+1 th step, the token value θ_j^h and θ_j^{h+1} of the place $p_j(j=1,2,3,\cdots,n,j\ne i)$ are totally the same, i.e., in θ_j^h and θ_j^{h+1}, the token values of the rest of places are exactly the same except the Sink places. The equivalent input of each transition is only connected with the token values of its input places, and it has nothing to do with the token values of its output places.

In conclusion, when $k = h$ and $k = h + 1$, the equivalent input ρ_h and ρ_{h+1} of each transition is equal. According to Equations (14)–(16), $\theta^{h+1} = \theta^h$, so the reasoning process has been finished according to Theorem 1.

(ii) Then, we prove $k < h+1$.
When $k = j, j < h+1$, if the equivalent input of each transition that is not fired is smaller than the threshold value of the corresponding transition, then $\rho_k' = \rho_k \oslash Th = (<0,1>, <0,1>, \cdots, <0,1>)^T$. Thus, the transition will not be fired anymore. According to Equations (14)–(16), θ^k does not change anymore, so when $k < h+1$, the reasoning process can be finished.

In conclusion, the theorem is proved.

Theorem 3 The computational complexity of the reasoning algorithm in the worst case is $O(nm^2)$.

Proof: Suppose no circuit exists in the IFPN model. In the general case, the computational complexity is $O(knm)$, where k is the circular times of the reasoning algorithm. Considering the worst case, i.e., the reasoning process proceeds $h + 1$ times (h is the transition number of the longest route in the IFPN), the computational complexity of the reasoning algorithm is $O((h+1)nm) = O((m+1)nm)$, that is $O(nm^2)$.

Therefore, in case there are n propositions and m rules in IFPR set S, and there are n places and m transitions in the corresponding IFPN model, the computational complexity of the reasoning algorithm based on the IFPN model is $O(nm^2)$.

4. Experiments and Results

In this section, a turbine fault diagnosis expert system [4] is used to prove the feasibility and the effectiveness of the proposed method. The turbine fault diagnosis expert system can be expressed as IFPR set S and the IFPN model of S shown in Figure. 5.

The IFPR set S contains the following IFPR:

R_1: IF *the cross section area of turbine's path is too large* (p_1)
AND *the efficiency of assembling unit is too low* (p_2)
THEN *the ventilation side of the guider's blade of turbine wears and tears* (p_3)(λ_1, CF_1)
R_2: IF *the spray head of turbine is broken* (p_{25})
AND *the flow rate of the fuel in the combustor is not too high* (p_{26})
THEN *the high pressure level's spray head of the turbine is broken* (p_8)(λ_2, CF_2)
R_3: IF *the spray head of turbine is broken* (p_{25})
AND *the flow rate of the fuel in the combustor is too high* (p_7)
THEN *the high pressure level's spray head of the turbine is broken* (p_8) (λ_3, CF_3)
R_4: IF *Outlet gas temperature of turbine is too high*(p_9)
AND *Efficiency of turbine is too low*(p_{10})
AND *Flow coefficient of turbine is too low* (p_{11})
THEN *Blade of the turbine scales*(p_{12})(λ_4, CF_4)
R_5: IF *Efficiency of turbine is too low*(p_{10})
AND *Power of assembling unit is too low* (p_{13})
AND *the efficiency of assembling unit is too low* (p_2)
THEN *Blade of the turbine wears and tears* (p_{14}) (λ_5, CF_5)
R_6: IF *the efficiency of assembling unit is too low* (p_2)
AND *Inlet gas temperature of turbine is too high* (p_{15})
THEN *Blade of the turbine burns down* (p_{16})(λ_6, CF_6)
R_7: IF *Flow path of compressor wears and tears*(p_{17})
THEN *Compressor is in turbulence* (p_{18}) (λ_7, CF_7)
R_8: IF *Blade of compress or breaks down* (p_{19})
THEN *Compressor is in turbulence* (p_{18}) (λ_8, CF_8)
R_9: IF *Conversion flow of the compressor is too low* (p_{20})
AND *Pressurization ratio of the compressor is too low* (p_5)
AND *Compressor has a problem* (p_{24})
THEN *Inlet of compressor freezes* (p_{22}) (λ_9, CF_9)
R_{10}: IF *the efficiency of assembling unit is too low* (p_2)
AND *Fuel consumption of assembling unit is too high* (p_{21})
AND *Inlet gas temperature of turbine is too high* (p_{15})
THEN *Compressor has a problem* (p_{24}) (λ_{10}, CF_{10})
R_{11}: IF *Pressurization ratio of the compressor is too low* (p_5)
AND *Uniform entropy compression efficiency of compressor is too low* (p_{23})
THEN *Blade of turbin scales*(p_{12})(λ_{11}, CF_{11})
R_{12}: IF *Conversion flow of the compressor is not too low* (p_{27})
AND *Pressurization ratio of the compressor is not too low* (p_{28})
AND *Compressor has a problem* (p_{24})
THEN *Flow path of compressor wears and tears*(p_{17}) (λ_{12}, CF_{12})
R_{13}: IF *Inlet gas temperature of turbine is too low* (p_4)
AND *the efficiency of assembling unit is too low* (p_2)
AND *Pressurization ratio of the compressor is too low* (p_5)
THEN *the spray head of turbine is broken* (p_{25}) (λ_{13}, CF_{13})

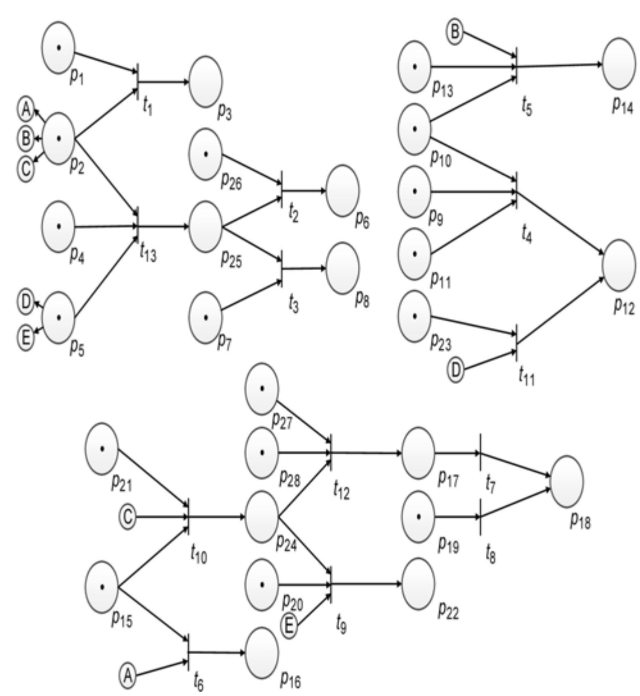


Figure 5. IFPN model of the turbine fault diagram expert system

According to the IFPN model and the IFPN definition, we can determine that the number of places is $n = 28$ and the number of transitions is $m = 13$. The input matrix I and output matrix O are as follows:

$$I=\begin{bmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&1&1&0&0&0&1&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&1&0&1&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&1&1&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&1&0&0&1&0\\
0&1&1&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0
\end{bmatrix}
\quad
O=\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}^T$$

Suppose

$$\begin{aligned}\theta^0 &= (\theta_1^0,\theta_2^0,\cdots,\theta_{28}^0)^T = (<\mu_1^0,\gamma_1^0>,<\mu_2^0,\gamma_2^0>,\cdots,<\mu_{28}^0,\gamma_{28}^0>)^T\\
&=(<0.6,0.3>,<0.6,0.3>,<0,1>,<0.5,0.4>,<0.7,0.2>,<0,1>,<0.7,0.2>,\\
&\quad <0,1>,<0.8,0.1>,<0.9,0.1>,<0.8,0.1>,<0,1>,<0.7,0.1>,<0,1>,\\
&\quad <0.5,0.2>,<0,1>,<0,1>,<0,1>,<0.7,0.2>,<0.7,0.2>,<0.7,0.2>,\\
&\quad <0,1>,<0.7,0.2>,<0,1>,<0,1>,<0.7,0.2>,<0.7,0.2>,<0.7,0.2>)^T\end{aligned}$$

$$\begin{aligned}Th &= (\lambda_1,\lambda_2,\cdots,\lambda_{13})^T\\
&=(<0.5,0.4>,<0.2,0.5>,<0.3,0.5>,<0.5,0.2>,<0.5,0.4>,\\
&\quad <0.4,0.3>,<0.5,0.4>,<0.3,0.5>,<0.6,0.2>,<0.4,0.3>,\\
&\quad <0.6,0.3>,<0.5,0.4>,<0.4,0.5>)^T\end{aligned}$$

$$\begin{aligned}CF &= diag(CF_1,CF_2,\cdots,CF_{13})\\
&= diag(<0.7,0.1>,<1,0>,<1,0>,<0.8,0.1>,<0.6,0.3>,<0.8,0.1>,\\
&\qquad <1,0>,<0.9,0.1>,<0.5,0.3>,<1,0>,<1,0>,<1,0>,<1,0>)\end{aligned}$$

$$\rho_0 = (\rho\theta_1^0,\rho\theta_2^0,\cdots,\rho\theta_{13}^0)^T = (<0,1>,<0,1>,\cdots,<0,1>)^T$$

The reasoning process based on IFPN is described as follows:

(1) Let $k=1$, and the reasoning results are as follows:
s

$$\begin{aligned}\rho_1 &= (\rho\theta_1^1,\rho\theta_2^1,\cdots,\rho\theta_{13}^1)^T = (<0.6,0.3>,<0,1>,<0,1>,<0.8,0.1>,<0.6,0.3>,\\
&\quad <0.5,0.3>,<0,1>,<0.7,0.2>,<0,1>,<0.5,0.3>,<0.7,0.2>,<0,1>,<0.5,0.4>)^T\end{aligned}$$

$$\begin{aligned}\rho_1' &= \rho_1 \oslash Th = (<0.6,0.3>,<0,1>,<0,1>,<0.8,0.1>,<0.6,0.3>,<0.5,0.3>,\\
&\quad <0,1>,<0.7,0.2>,<0,1>,<0.5,0.3>,<0.7,0.2>,<0,1>,<0.5,0.4>)^T\end{aligned}$$

$$\begin{aligned}\theta^1 &= (<0.6,0.3>,<0.6,0.3>,<0.42,0.37>,<0.5,0.4>,<0.7,0.2>,<0,1>,\\
&\quad <0.7,0.2>,<0,1>,<0.8,0.1>,<0.9,0.1>,<0.8,0.1>,<0.64,0.2>,\\
&\quad <0.7,0.1>,<0.36,0.51>,<0.5,0.2>,<0.4,0.37>,<0,1>,<0.63,0.28>,\\
&\quad <0.7,0.2>,<0.7,0.2>,<0.7,0.2>,<0,1>,<0.7,0.2>,<0.5,0.3>,\\
&\quad <0.5,0.4>,<0.7,0.2>,<0.7,0.2>,<0.7,0.2>)^T\end{aligned}$$

Given that $\theta^1 \neq \theta^0$, continue the reasoning process.

(2) Let $k=2$, and the reasoning results are as follows:

$$\begin{aligned}\rho_2 &= (\rho\theta_1^2,\rho\theta_2^2,\cdots,\rho\theta_{13}^2)^T = (<0.6,0.3>,<0.5,0.4>,<0.5,0.4>,<0.8,0.1>,<0.6,0.3>,\\
&\quad <0.5,0.3>,<0,1>,<0.7,0.2>,<0.5,0.3>,<0.5,0.3>,<0.7,0.2>,<0.5,0.3>,<0.5,0.4>)^T\end{aligned}$$

$$\begin{aligned}\rho_2' &= \rho_2 \oslash Th = (<0.6,0.3>,<0.5,0.4>,<0.5,0.4>,<0.8,0.1>,<0.6,0.3>,\\
&\qquad <0.5,0.3>,<0,1>,<0.7,0.2>,<0,1>,<0.5,0.3>,<0.7,0.2>,\\
&\qquad <0.5,0.3>,<0.5,0.4>)^T\end{aligned}$$

$$\begin{aligned}\theta^2 &= (<0.6,0.3>,<0.6,0.3>,<0.42,0.37>,<0.5,0.4>,<0.7,0.2>,<0.5,0.4>,\\
&\quad <0.7,0.2>,<0.5,0.4>,<0.8,0.1>,<0.9,0.1>,<0.8,0.1>,<0.64,0.2>,\\
&\quad <0.7,0.1>,<0.36,0.51>,<0.5,0.2>,<0.4,0.37>,<0.5,0.3>,<0.63,0.28>,\\
&\quad <0.7,0.2>,<0.7,0.2>,<0.7,0.2>,<0,1>,<0.7,0.2>,<0.5,0.3>,<0.5,0.4>,\\
&\quad <0.7,0.2>,<0.7,0.2>,<0.7,0.2>)^T\end{aligned}$$

Given that $\theta^2 \neq \theta^1$, continue the reasoning process.

(3) Let $k=3$, and the reasoning results are as follows:

$$\rho_3 = (\rho\theta_1^3, \rho\theta_2^3, \cdots, \rho\theta_{13}^3)^T$$
$$= (<0.6, 0.3>, <0.5, 0.4>, <0.5, 0.4>, <0.8, 0.1>, <0.6, 0.3>, <0.5, 0.3>, <0.5, 0.3>, <0.7, 0.2>, <0.5, 0.3>, <0.5, 0.3>, <0.7, 0.2>, <0.5, 0.3>, <0.5, 0.4>)^T$$

$$\rho_3' = \rho_3 \oslash Th$$
$$= (<0.6, 0.3>, <0.5, 0.4>, <0.5, 0.4>, <0.8, 0.1>, <0.6, 0.3>, <0.5, 0.3>, <0.5, 0.3>, <0.7, 0.2>, <0, 1>, <0.5, 0.3>, <0.7, 0.2>, <0.5, 0.3>, <0.5, 0.4>)^T$$

$$\theta^3 = (<0.6, 0.3>, <0.6, 0.3>, <0.42, 0.37>, <0.5, 0.4>, <0.7, 0.2>, <0.5, 0.4>, <0.7, 0.2>, <0.5, 0.4>, <0.8, 0.1>, <0.9, 0.1>, <0.8, 0.1>, <0.64, 0.2>, <0.7, 0.1>, <0.36, 0.51>, <0.5, 0.2>, <0.4, 0.37>, <0.5, 0.3>, <0.5, 0.3>, <0.7, 0.2>, <0.7, 0.2>, <0.7, 0.2>, <0, 1>, <0.7, 0.2>, <0.5, 0.3>, <0.5, 0.4>, <0.7, 0.2>, <0.7, 0.2>, <0.7, 0.2>)^T$$

Given that $\theta^3 \neq \theta^2$, continue the reasoning process.

(4) Let $k = 4$, and the reasoning results are as follows:

$$\theta^4 = (<0.6, 0.3>, <0.6, 0.3>, <0.42, 0.37>, <0.5, 0.4>, <0.7, 0.2>, <0.5, 0.4>, <0.7, 0.2>, <0.5, 0.4>, <0.8, 0.1>, <0.9, 0.1>, <0.8, 0.1>, <0.64, 0.2>, <0.7, 0.1>, <0.36, 0.51>, <0.5, 0.2>, <0.4, 0.37>, <0.5, 0.3>, <0.5, 0.3>, <0.7, 0.2>, <0.7, 0.2>, <0.7, 0.2>, <0, 1>, <0.7, 0.2>, <0.5, 0.3>, <0.5, 0.4>, <0.7, 0.2>, <0.7, 0.2>, <0.7, 0.2>)^T$$

As $\theta^4 = \theta^3$, end the reasoning process.

After completing the reasoning process, output the final truth degree of all positions θ^4 and the times of iteration $k = 4$. We can obtain the token value of place p_{18} from θ^4, and its value is $<0.5, 0.3>$.

The value indicates that the truth of proposition "*Compressor is in turbulence*" is $<0.5, 0.3>$; 0.5 is its membership degree , 0.3 is nonmembership degree, and 0.2 = 1- 0.5 - 0.3 is its hesitancy degree. These results show that the possibility of the fault "*Compressor is in turbulence*" is 0.5, its impossibility is 0.3, and its uncertainty is 0.2.

If we use the reasoning algorithm based on FPN to compute the truth of proposition "*Compressor is in turbulence,*" as proposed in [2][3][4][5][6], we can only obtain a single membership. The reasoning result only shows that the possibility and the impossibility of the fault "***C****ompressor is in turbulence"* are 0.5, and the uncertainty of the fault "*Compressor is in turbulence*" is unknown.

When we compared the reasoning result obtained from FPN with that obtained from IFPN, we found that the reasoning result obtained from IFPN was more comprehensive and precise and it could describe the uncertainty state (i.e., neutral state).

In addition, we can infer that the reasoning process only circulates four times; it is carried out by matrix operation and is a parallel computing process. Compared with the reasoning algorithm proposed in [2][3], the reasoning algorithm proposed in this paper is simple, efficient, and can make full use of parallel computing capability of Petri nets.

5. Conclusion

In this paper, an IFPN model for knowledge representation and reasoning was proposed to solve the single membership issue of the FPN model. The IFPN model was constructed by combining IFS theory and Petri nets theory, and the reasoning process based on IFPN was carried out by matrix operation. The main conclusions of this paper are as follows:

(1) The IFPN model can overcome the single membership degree limitation of the FPN model. It can also describe the neutral state and the reasoning result more comprehensively and precisely because it increases a nonmembership function.

(2) The reasoning process based on IFPN is a parallel computing process, so it is simple, efficient, and can make full use of the parallel computing capability of Petri nets.

(3) IFPN is a promising knowledge representation method, and it is an effective extension and development of FPN. IFPN can be used for modeling and solving uncertainty problems, such as fault diagnosis, exception handling, and decision making.

(4) Further research based on IFPN is necessary, especially for the learning capacity of IFPN, because the IFPN has many parameters, such as threshold value and certainty factor.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (Grant No. 61272011).

References

[1] Liu, H. C., Liu, L., Lin, Q. L., et al (2013). Knowledge acquisition and representation using fuzzy evidential reasoning and dynamic adaptive fuzzy Petri nets. *IEEE Transactions on Cybernetics,* 43 (3) 1059-1072.

[2] Chen, S. M., Ke, J. S., Chang, J. F. (1990). Knowledge representation using fuzzy Petri nets. *IEEE Transaction on Knowledge and Data Engineering,* 2 (3) 311-319.

[3]Chen, S. M. (2002). Weighted fuzzy reasoning using weighted fuzzy Petri nets. *IEEE Transaction on Knowledge and Data Engineering,* 14 (2) 386-397.

[4] Gao, M. M., Zhou, M. C., Huang, X. G., Wu. Z. M.(2003). Fuzzy reasoning Petri nets. I*EEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans,* 33 (3) 314-324.

[5] Wang, Y., Ling, C., Qu, Y., et al (2003). Consistent Fuzzy Petri Nets Model for Logic Programs with Negation. *Acta Electronica Sinica*, 34 (11) 1955-1960.

[6] Jia, L. X., Xun, J. Y., Ru, F. (2003). Fuzzy Petri Net Based Formalized Reasoning Algorithm with Applications. *Journal of Xi'an Jiaotong University,* 37 (12) 1263-1266.

[7] Li, X. O., Yu, W., Lara-Rosano, F. (2000). Dynamic knowledge inference and learning under adaptive fuzzy Petri net framework. I*EEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 30 (4) 442-450.

[8] Liu, H. C., Lin, Q. L,. Mao, L. X., et al(2013). Dynamic Adaptive Fuzzy Petri Nets for Knowledge Representation and Reasoning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* 43 (6) 1399-1410.

[9] Shen, V. (2006). Knowledge Representation Using High-Level Fuzzy Petri Nets. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans,* 36 (6) 2120-2127.

[10] Ribaric, S., Hrkac, T. (2012). A model of fuzzy spatio-temporal knowledge representation and reasoning based on high-level Petri nets. *Information Systems*, 37 (3) 238-256.

[11] Amin, M., Shebl, D. (2014). Reasoning dynamic fuzzy systems based on adaptive fuzzy higher order Petri nets. *Information Sciences,* 286 (10) 161–172.

[12]Pedrycz, W., Camargo, H. (2003). Fuzzy timed Petri nets. *Fuzzy Sets and Systems,* 149 (2) 301-330.

[13] Liu, Z., Li, H., Zhou, P. (2011). Towards timed fuzzy Petri net algorithms for chemical abnormality monitoring.*Expert Systems with Applications,* 38 (8) 9724-9728.

[14]Wang, X., Zhang, L., Dou, W., et al. (2013). Fuzzy colored time Petri net and termination analysis for fuzzy Event-Condition-Action rules. *Information Sciences,* 232 (5) 225-240.

[15] Chiang, W., Liu, K., Lee, J. (2014). Bridge Damage Assessment through Fuzzy Petri Net Based Expert System. *Journal of Computing in Civil Engineering*, 14 (2) 141-149.

[16]Hu, H., Li, Z., Al-Ahmari, A. (2011). Reversed fuzzy Petri nets and their application for fault diagnosis. *Computers & Industrial Engineering,* 60 (4) 505-510.

[17]Liu, H., Lin, Q., Ren, M. (2013). Fault diagnosis and cause analysis using fuzzy evidential reasoning approach and dynamic adaptive fuzzy Petri nets. *Computers & Industrial Engineering,* 66 (4), 899-908.

[18]Gniewek, L. (2013). Sequential control algorithm in the form of fuzzy interpreted Petri net. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43 (2), 451-459.

[19]Lin, C. M., Li, H. Y. (2015). Dynamic Petri Fuzzy Cerebellar Model Articulation Controller Design for a Magnetic Levitation System and a Two-Axis Linear Piezoelectric Ceramic Motor Drive System. *IEEE Transactions on Control Systems Technology,* 23 (2), 693-699.

[20]Ye, Y., Jiang, Z., Diao, X., et al (2011). Extended event–condition–action rules and fuzzy Petri nets based exception handling for workflow management. *Expert Systems with Applications,* 38 (9), 10847-10861.

[21] Atanassov, K (1986). Intuitionistic fuzzy sets. *Fuzzy Sets and Systems,* 20 (1) 87-96.