

# 基于直觉模糊 Petri 网的混合推理方法

孟飞翔<sup>1</sup>, 雷英杰<sup>1</sup>, 雷 阳<sup>2</sup>, 申晓勇<sup>1</sup>

(1. 空军工程大学防空反导学院, 陕西西安 710051; 2. 武警工程大学, 陕西西安 710086)

**摘 要:** 针对现有的基于模糊 Petri 网(Fuzzy Petri Nets, FPN)和直觉模糊 Petri 网(Intuitionistic Fuzzy Petri Nets, IFPN)的推理方法在求解只涉及知识库中部分规则的问题时存在推理过程复杂、效率不高, 而且不能对问题产生的原因进行分析等缺陷, 提出一种基于 IFPN 的混合推理方法. 该方法将反向推理与正向推理相结合, 首先把所要求解的问题转化为目标库所, 并引入关联库所、关联变迁和子模型等概念; 其次运用反向推理寻找目标库所的关联库所和变迁并构建推理子模型, 从而获取问题产生的潜在原因并简化推理模型; 最后以子模型作为推理模型, 运用正向推理求解目标库所的 token 值, 解决了直接运用原模型进行推理时过程复杂且效率不高的问题. 与此同时, 通过在模型中引入阈值以及“路径”和“有效路径”等定义, 排除无效关联库所, 从而找出了问题产生的真正原因. 实例验证表明该方法可行且有效, 与现有方法的对比分析表明该方法克服了现有方法的缺陷.

**关键词:** 直觉模糊 Petri 网; 正向推理; 反向推理; 混合推理

**中图分类号:** TP18      **文献标识码:** A      **文章编号:** 0372-2112 (2017)08-1937-10

**电子学报 URL:** <http://www.ejournal.org.cn>      **DOI:** 10.3969/j.issn.0372-2112.2017.08.019

## Hybrid Reasoning Using Intuitionistic Fuzzy Petri Nets

MENG Fei-xiang<sup>1</sup>, LEI Ying-jie<sup>1</sup>, LEI Yang<sup>2</sup>, SHEN Xiao-yong<sup>1</sup>

(1. Air and Missile Defense College, Air Force Engineering University, Xi'an, Shaanxi 710051, China;

2. Engineering University of Armed Police Force, Xi'an, Shaanxi 710086, China)

**Abstract:** Aimed at that existing reasoning methods based on fuzzy Petri nets (FPN) and intuitionistic fuzzy Petri nets (IFPN) have the defects of complicated process and low efficiency in solving the problems only related to part of the knowledge base and they cannot analyze the causes of the problems, a hybrid reasoning method based on IFPN was presented. The method combined the forward reasoning and backward reasoning, firstly, the problems which needed to be solved were converted to goal places; secondly, in order to obtain potential causes of the problems and to simplify the reasoning model, associate places and transitions of goal places were searched by backward reasoning and a submodel was constructed; lastly, the problems of complicated process and low efficiency in using the original model to reason were solved by taking the submodel as the reasoning model and using forward reasoning to compute token values of goal places. Moreover, the real causes of the problems were found out by introducing threshold and the definition of route and active route into the model to remove the invalid associate places. The examples shows that the hybrid reasoning method is feasible and effective, and that compared with the existing methods shows that it overcomes the defects of the existing methods.

**Key words:** intuitionistic fuzzy Petri nets (IFPN); forward reasoning; backward reasoning; hybrid reasoning

## 1 引言

FPN 和 IFPN 由于综合了 Petri 网的图形描述能力和模糊系统的推理能力而被广泛用于知识的表示和推理<sup>[1-5]</sup>、建模仿真<sup>[6]</sup>、异常检测<sup>[14]</sup>以及故障诊断<sup>[15,16]</sup>等领域. 近年来, 国内外学者对基于 FPN 和 IFPN 的推理方法进行了深入的研究. Chen<sup>[1]</sup>首先对基于 FPN 的推理算法进行了探索; 接着 Gao 等<sup>[2]</sup>提出了基于模糊推理 Petri 网

的推理算法; 汪洋等<sup>[3]</sup>提出了基于一致性模糊 Petri 网的推理算法; Li 等<sup>[4]</sup>研究了基于自适应模糊 Petri 网的推理算法; Liu 等<sup>[5]</sup>提出了基于动态自适应模糊 Petri 网的推理算法. 与此同时, 针对 FPN 存在隶属度单一的缺陷, 一些学者将直觉模糊理论与 Petri 网理论相结合, 解决了这一问题. Shen 等<sup>[7]</sup>对 IFPN 推理模型和算法进行了探索; Liu 等<sup>[8]</sup>将直觉模糊理论引入 FPN 模型中; 孟飞翔等<sup>[9]</sup>提出了基于矩阵运算的 IFPN 推理方法.

上述研究极大地推动了 FPN 和 IFPN 的发展和运用,这些推理方法通常是从已知条件出发按照某种策略求解问题,可统称为正向推理(forward reasoning),但在实际运用中它们却存在如下两个问题:一是由于上述推理方法运用的推理模型大多建立在整个知识库上,所以每次推理时,都需要使用知识库中的所有规则.而在实际运用过程中,当对单个或者部分问题进行分析时(例如在文献[10]中判断二手车的价格处于中等价位的可信度),仅需使用知识库中的部分规则即可.此时,如果继续采用上述推理方法,当整个知识库特别庞大时,推理过程不仅复杂而且推理效率较低.二是上述推理方法是运用已知条件求解问题的过程,属于数据驱动的方式,所以在推理结束后无法对问题产生的原因进行分析.例如现有用于故障诊断领域的基于 FPN 的推理方法,在根据已知事实推理得出可能出现的故障后,却无法对每个故障产生的原因进行进一步的分析,这不利于故障的诊断和预防.

针对第一个问题,Scarpell 等<sup>[11]</sup>提出了基于高层次模糊 Petri 网的反向搜索算法,Chen<sup>[10]</sup>提出了基于模糊与或图的 FPN 反向推理算法,这两种方法都是以推理模型为基础,从目标库所出发,沿着有向弧逐步搜索目标库所的关联库所和变迁并构建关联 Petri 网模型,虽然可以简化模型,但本质上都是基于图形的搜索算法,算法并没有充分利用 Petri 网的并行推理功能,而且当图形结构十分复杂时,算法的搜索效率就会十分低下;鲍培明<sup>[12]</sup>提出了一种建立在 FPN 的基本结构上的反向推理算法,算法简单高效,但是由于推理模型中缺乏阈值,因此无法从关联库所中筛选出目标库所的有效关联库所;Yuan 等<sup>[13]</sup>指出文献[12]提出的方法过于抽象而且未充分利用 FPN 的结构,于是在反向推理过程中引入中间库所和人机交互环节,在已知中间库所的 token 值的条件下,该算法能有效提高推理效率,但是现有的 FPN 推理方法中,已知的都是初始库所的 token 值,中间库所的 token 值是在推理过程中产生的,所以该算法的适用范围具有一定的局限性.针对第二个问题,Ye 等<sup>[14]</sup>提出基于 FPN 的工作流异常处理方法,在推理过程中同时使用了正向推理和反向推理,该算法的反向推理是基于图形的推理算法,算法虽然直观但是不便于和正向推理相结合;Hu 等<sup>[15]</sup>和 Liu 等<sup>[16]</sup>分别提出了基于逆向模糊 Petri 网和逆向动态自适应模糊 Petri 网的故障分析方法,但是这两种方法不能对单个故障产生的原因进行分析,而且文献[15]由于在推理模型中未考虑变迁的阈值,因此无法排除无效的故障原因.

上述研究主要集中于 FPN 领域,而针对 IFPN 领域存在的这些问题尚未开展实质性的研究,另外考虑到 IFPN 是 FPN 的有效扩充和发展,并成功克服了 FPN 隶属度单一的缺陷<sup>[9]</sup>,于是本文提出一种基于 IFPN 的混

合推理方法.该方法将反向推理与正向推理相结合,首先把所要求解的问题转化为 IFPN 推理模型中的目标库所,并引入关联库所、关联变迁和子模型等概念;接着运用反向推理寻找目标库所的关联库所和变迁从而获取问题产生的潜在原因;然后根据关联库所和变迁构建子模型,从而将原模型转化为一个相对简单且仅与目标库所相关联的子模型;最后以子模型为推理模型,运用正向推理求解目标库所的 token 值,解决了运用原模型推理过程复杂且效率不高的问题.为充分利用 Petri 网的并行运算能力并克服文献[10]和[11]中基于图形的搜索算法效率不高的缺陷,正反向推理都采用矩阵运算;同时通过将需要求解的问题转化为目标库所并在模型中引入“阈值”、“路径”和“有效路径”等定义,解决了文献[14~16]不能对单个问题产生的原因进行分析以及无法排除无效原因的问题.

## 2 基于 IFPN 的推理规则表示方法

### 2.1 IFPN 的定义

定义1(直觉模糊 Petri 网)<sup>[9]</sup> IFPN 可定义为一个 8 元组,  $IFPN = (P, T, F; I, O, \theta, Th, CF)$ , 其中:

- (1)  $P = \{p_1, p_2, \dots, p_n\}$  是一个有限库所集合;
- (2)  $T = \{t_1, t_2, \dots, t_m\}$  是一个有限变迁集合;
- (3)  $F \subseteq (P \times T) \cup (T \times P)$  是一个有向弧集合;
- (4)  $I: P \times T \rightarrow \{0, 1\}$  是一个表示从库所到变迁的  $n \times m$  维输入转移矩阵;
- (5)  $O: T \times P \rightarrow \{0, 1\}$  是一个表示从变迁到库所的  $m \times n$  维输出转移矩阵;
- (6)  $\theta = (\theta_1, \theta_2, \dots, \theta_n)^T$  表示库所的 token 值;
- (7)  $Th = (\lambda_1, \lambda_2, \dots, \lambda_m)^T$  表示变迁的阈值;
- (8)  $CF = \text{diag}(CF_1, CF_2, \dots, CF_m)$  表示所有变迁的可信度.

### 2.2 基于 IFPN 的 IFPR 表示方法

本文将常见的直觉模糊产生式规则(Intuitionistic Fuzzy Production Rules, IFPR)归纳为以下 4 种<sup>[9]</sup>,并根据文献[9]中的 IFPR 集与 IFPN 模型的对应关系分别给出每种类型所对应的 IFPN 模型、变迁触发条件以及变迁触发后库所的 token 值传递规则.

- (1) 简单的 IFPR 及其 IFPN 模型(如图 1)

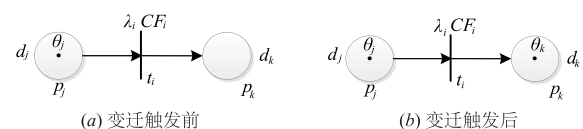


图1 简单的IFPR的IFPN模型

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

假设  $\theta_j = \langle \mu_j, \gamma_j \rangle$ ,  $\theta_k = \langle \mu_k, \gamma_k \rangle$ ,  $\lambda_i = \langle \alpha_i, \beta_i$

$\mu_j, \gamma_j, \lambda_i = \langle \alpha_i, \beta_i \rangle, CF_i = \langle C\mu_i, C\gamma_i \rangle$ , 则

①变迁的触发条件

当且仅当同时满足  $\begin{cases} \mu_j \geq \alpha \\ \gamma_j \leq \beta \end{cases}$  时, 变迁  $t_i$  才能触发;

②变迁触发后库所的 token 值传递规则

$$\begin{cases} \mu_k = \mu_j \times C\mu_i \\ \gamma_k = \gamma_j + C\gamma_i - \gamma_j \times C\gamma_i \end{cases} \quad (1)$$

(2) 具有合取式前提条件的 IFPR 及其 IFPN 模型 (如图 2)

$R_i: \text{IF } d_{j_1} \text{ AND } d_{j_2} \text{ AND } \dots \text{ AND } d_{j_m} \text{ THEN } d_k (CF_i, \lambda_i)$

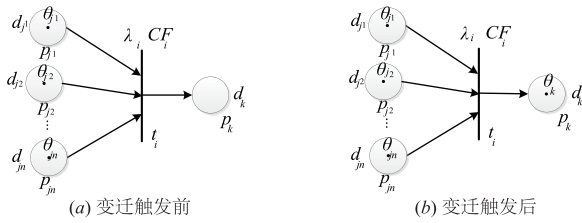


图2 具有合取式前提条件的IFPR的IFPN模型

假设  $\theta_{j_m} = \langle \mu_{j_m}, \gamma_{j_m} \rangle (m = 1, 2, \dots, n)$ ,  $\theta_k = \langle \mu_k, \gamma_k \rangle$ ,  $\lambda_i = \langle \alpha_i, \beta_i \rangle$ ,  $CF_i = \langle C\mu_i, C\gamma_i \rangle$ , 则

①变迁的触发条件

当且仅当同时满足  $\begin{cases} \min(\mu_{j_1}, \mu_{j_2}, \dots, \mu_{j_m}) \geq \alpha_i \\ \max(\gamma_{j_1}, \gamma_{j_2}, \dots, \gamma_{j_m}) \leq \beta_i \end{cases}$  时,

变迁  $t_i$  才能触发;

②变迁触发后库所的 token 值传递规则

$$\begin{cases} \mu_k = \min(\mu_{j_1}, \mu_{j_2}, \dots, \mu_{j_m}) \times C\mu_i \\ \gamma_k = \max(\gamma_{j_1}, \gamma_{j_2}, \dots, \gamma_{j_m}) + C\gamma_i - \max(\gamma_{j_1}, \gamma_{j_2}, \dots, \gamma_{j_m}) \times C\gamma_i \end{cases} \quad (2)$$

(3) 具有析取式前提条件的 IFPR 及其 IFPN 模型 (如图 3)

$R_i: \text{IF } d_{j_1} \text{ OR } d_{j_2} \text{ OR } \dots \text{ OR } d_{j_m} \text{ THEN } d_k (CF_i, \lambda_i)$

可以等价如下  $n$  条规则

$R_{i1}: \text{IF } d_{j_1} \text{ THEN } d_k (CF_i, \lambda_i)$

$R_{i2}: \text{IF } d_{j_2} \text{ THEN } d_k (CF_i, \lambda_i)$

.....

$R_{in}: \text{IF } d_{j_n} \text{ THEN } d_k (CF_i, \lambda_i)$

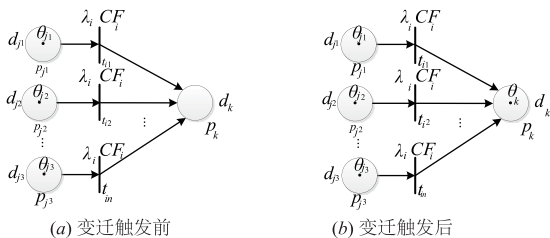


图3 具有析取式前提条件的IFPR的IFPN模型

假设  $\theta_{j_m} = \langle \mu_{j_m}, \gamma_{j_m} \rangle (m = 1, 2, \dots, n)$  和  $\theta_k = \langle$

$\mu_k, \gamma_k \rangle, \lambda_i = \langle \alpha_i, \beta_i \rangle, CF_i = \langle C\mu_i, C\gamma_i \rangle$ , 则

①变迁的触发条件

库所  $p_{j_1}, p_{j_2}, \dots, p_{j_m}$  中至少有  $m$  个库所的 token 值同

时满足  $\begin{cases} \mu_{j_m} \geq \alpha_i \\ \gamma_{j_m} \leq \beta_i \end{cases}$ , 变迁  $t_i$  才能触发;

②变迁触发后库所的 token 值传递规则

$$\begin{cases} \mu_k = \max(\mu_{j_1} \times C\mu_i, \mu_{j_2} \times C\mu_i, \dots, \mu_{j_m} \times C\mu_i) \\ \gamma_k = \min(\gamma_{j_1} + C\gamma_i - \gamma_{j_1} \times C\gamma_i, \gamma_{j_2} + C\gamma_i - \gamma_{j_2} \times C\gamma_i, \dots, \gamma_{j_m} + C\gamma_i - \gamma_{j_m} \times C\gamma_i) \end{cases} \quad (3)$$

(4) 具有合取式结论的 IFPR 及其 IFPN 模型 (如图 4)

4)

$R_i: \text{IF } d_j \text{ THEN } d_{k_1} \text{ AND } d_{k_2} \text{ AND } \dots \text{ AND } d_{k_n} (CF_i, \lambda_i)$

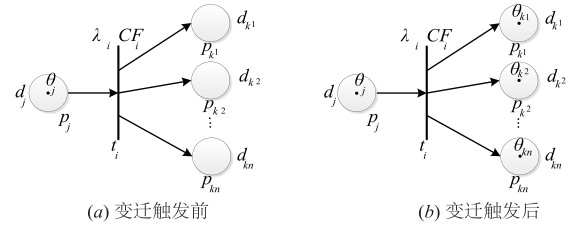


图4 具有合取式结论的IFPR的IFPN模型

假设  $\theta_j = \langle \mu_j, \gamma_j \rangle$ ,  $\theta_{k_i} = \langle \mu_{k_i}, \gamma_{k_i} \rangle (i = 1, 2, \dots, n)$ ,  $\lambda_i = \langle \alpha_i, \beta_i \rangle$ ,  $CF_i = \langle C\mu_i, C\gamma_i \rangle$

①变迁的触发条件

当且仅当同时满足  $\begin{cases} \mu_j \geq \alpha_i \\ \gamma_j \leq \beta_i \end{cases}$  时, 变迁  $t_i$  才能触发;

②变迁触发后库所的 token 值传递规则

$$\begin{cases} \mu_{k_i} = \mu_j \times C\mu_i \\ \gamma_{k_i} = \gamma_j + C\gamma_i - \gamma_j \times C\gamma_i \end{cases} \quad (4)$$

### 3 基于 IFPN 的混合推理方法

#### 3.1 混合推理方法

##### 3.1.1 相关定义

定义 2 (前集, 后集) 设 IFPN =  $(P, T, F; I, O, \theta, Th, CF)$  为一直觉模糊 Petri 网, 称

$\cdot p = \{t \mid (t, p) \in F\}$  为库所  $p$  的前集 (pre-set);

$p \cdot = \{t \mid (p, t) \in F\}$  为库所  $p$  的后集 (post-set);

$\cdot t = \{p \mid (p, t) \in F\}$  为变迁  $t$  的前集 (pre-set);

$t \cdot = \{p \mid (t, p) \in F\}$  为变迁  $t$  的后集 (post-set).

定义 3 (直接可达集, 可达集<sup>[11]</sup>) 在 IFPN 中, 如果  $p_i \in \cdot t_i, p_j \in t_i \cdot$ , 则称从  $p_i$  直接可达  $p_j$ , 记为  $p_i \Rightarrow p_j$ ; 如果  $p_i \Rightarrow p_j, p_j \Rightarrow p_{j+1}, \dots, p_{j+k-1} \Rightarrow p_{j+k}$ , 则称从  $p_i$  可达  $p_{j+1}, p_{j+2}, \dots, p_{j+k}$ , 记为  $p_i \rightarrow p_{j+1}, p_i \rightarrow p_{j+2}, \dots, p_i \rightarrow p_{j+k}$ . 所有从  $p_i$  直接可达的库所构成的集合称为  $p_i$  的直接可达集 (immediate reachability set), 记为  $IRS(p_i)$ ; 所有从  $p_i$  可达的库所构成的集合称为  $p_i$  的可达集 (reachability set), 记为  $RS(p_i)$ .

定义 4 (反向直接可达集, 反向可达集) 在 IFPN

中,如果从  $p_i$  直接可达  $p_j$ ,则称从  $p_j$  反向直接可达  $p_i$ ;如果从  $p_i$  可达  $p_k$ ,则称从  $p_k$  反向可达  $p_i$ .所有从  $p_j$  反向直接可达的库所构成的集合称为  $p_j$  的反向直接可达集(backward immediate reachability set),记为  $BIRS(p_j)$ ;所有从  $p_k$  反向可达的库所构成的集合称为  $p_k$  的反向可达集(backward reachability set),记为  $BRS(p_k)$ .

**定义 5**(源库所,终结库所<sup>[4]</sup>) 在 IFPN 中,若一个库所没有输入变迁,则称该库所为源库所(Source Places);若一个库所没有输出变迁,则称该库所为终结库所(Sink Places).

**定义 6**(关联库所集,关联变迁集) 在 IFPN 中,假设  $p_i$  为源库所, $p_g$  为目标库所,所有变迁都能触发,如果  $p_g$  能通过  $p_1, p_2, \dots, p_j$  和  $t_1, t_2, \dots, t_j$  顺序地从  $p_i$  中获取 token 值,则称  $p_i$  和  $p_1, p_2, \dots, p_j$  为  $p_g$  的关联库所, $t_1, t_2, \dots, t_j$  为  $p_g$  的关联变迁. $p_g$  的所有关联库所构成的集合称为  $p_g$  的关联库所集(incident places set),记为  $IPS(p_g)$ ;  $p_g$  的所有关联变迁构成的集合称为  $p_g$  的关联变迁集(incident transitions set),记为  $ITS(p_g)$ .

**定义 7**(子网) 设  $IFPN = (P, T, F; I, O, \theta, Th, CF)$  为一直觉模糊 Petri 网,如果满足如下条件:

- (1)  $P' \subseteq P$ ;
- (2)  $T' \subseteq T$ ;
- (3)  $F' = ((P' \times T') \cup (T' \times P')) \cap F$ ;

则称  $S\text{-IFPN} = (P', T', F'; I', O', \theta', Th', CF')$  为该直觉模糊 Petri 网的子网或子模型.

### 3.1.2 定义算子

为简洁地表示推理算法,定义如下算子:

- (1) 加法算子  $\hat{+}$

$$C = A \hat{+} B,$$

其中  $C = (c_{ij})_{m \times n}$ ,  $A = (a_{ij})_{m \times n}$ ,  $B = (b_{ij})_{m \times n}$ ,

$$c_{ij} = \max(a_{ij}, b_{ij});$$

- (2) 加法算子  $\hat{\oplus}$

$$C = A \hat{\oplus} B,$$

其中  $C = (c_{ij})_{m \times n} = (\langle \mu c_{ij}, \gamma c_{ij} \rangle)_{m \times n}$ ,

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

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

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

- (3) 乘法算子  $\hat{*}$

$$C = A \hat{*} B,$$

其中  $C = (c_{ij})_{m \times l}$ ,  $A = (a_{ij})_{m \times n}$ ,  $B = (b_{ij})_{n \times l}$ ,

$$c_{ij} = \max_{1 \leq k \leq n} (a_{ik} \times b_{kj})$$

- (4) 乘法算子  $\hat{\otimes}$

$$C = A \hat{\otimes} B,$$

其中  $C = (c_i)_{n \times 1} = (\langle c\mu_i, c\gamma_i \rangle)_{n \times 1}$ ,

$$A = (a_{ij})_{n \times m},$$

$$B = (b_j)_{m \times 1} = (\langle b\mu_j, b\gamma_j \rangle)_{m \times 1},$$

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

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

- (5) 比较算子  $\hat{\odot}$

$$C = A \hat{\odot} B,$$

其中

$$C = (c_1, c_2, \dots, c_m)^T = (\langle \mu c_1, \gamma c_1 \rangle, \langle \mu c_2, \gamma c_2 \rangle, \dots, \langle \mu c_m, \gamma c_m \rangle)^T,$$

$$A = (a_1, a_2, \dots, a_m)^T = (\langle \mu a_1, \gamma a_1 \rangle, \langle \mu a_2, \gamma a_2 \rangle, \dots, \langle \mu a_m, \gamma a_m \rangle)^T,$$

$$B = (b_1, b_2, \dots, b_m)^T = (\langle \mu b_1, \gamma b_1 \rangle, \langle \mu b_2, \gamma b_2 \rangle, \dots, \langle \mu b_m, \gamma b_m \rangle)^T,$$

$$c_i = \langle \mu c_i, \gamma c_i \rangle = \begin{cases} \langle \mu a_i, \gamma a_i \rangle, \mu c_i \geq \mu a_i \text{ and } \gamma c_i \leq \gamma a_i \\ (0, 1), \text{ other} \end{cases}$$

- (6) 直乘算子  $\hat{\odot}$

$$C = A \hat{\odot} B,$$

其中  $C = (c_{ij})_{m \times n} = (\langle \mu c_{ij}, \gamma c_{ij} \rangle)_{m \times n}$ ,

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

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

$$c_{ij} = \langle \mu c_{ij}, \gamma c_{ij} \rangle$$

$$= \langle \mu a_{ij} \times \mu b_{ij}, \gamma a_{ij} + \gamma b_{ij} - \gamma a_{ij} \times \gamma b_{ij} \rangle$$

- (7) 向量否定算子  $neg$

已知  $\theta = (\theta_1, \theta_2, \dots, \theta_n)^T = (\langle \mu_1, \gamma_1 \rangle, \langle \mu_2, \gamma_2 \rangle, \dots, \langle \mu_n, \gamma_n \rangle)^T$ , 则

$$neg\theta = neg(\theta_1, \theta_2, \dots, \theta_n)^T$$

$$= neg(\langle \mu_1, \gamma_1 \rangle, \langle \mu_2, \gamma_2 \rangle, \dots, \langle \mu_n, \gamma_n \rangle)^T$$

$$= (\langle \gamma_1, \mu_1 \rangle, \langle \gamma_2, \mu_2 \rangle, \dots, \langle \gamma_n, \mu_n \rangle)^T = \bar{\theta}$$

### 3.1.3 定义向量

- (1) 关联库所向量  $AP$

$AP = (x_0, x_1, \dots, x_n)^T$ , 若  $p_i$  是目标库所或目标库所的关联库所, 则  $x_i = 1$ ; 否则  $x_i = 0$ ;

- (2) 关联变迁向量  $AT$

$AT = (y_0, y_1, \dots, y_m)^T$ , 若  $t_j$  是目标库所的关联变迁, 则  $y_j = 1$ ; 否则  $y_j = 0$ .

### 3.1.4 混合推理方法

基于 IFPN 的混合推理方法分为两部分: 构建推理子模型部分和求解目标库所 token 值部分. 该方法首先运用反向推理获取目标库所的关联库所集和关联变迁集并构建子模型, 然后以子模型作为推理模型, 运用正向推理求解目标库所的 token 值.

在大多数实际应用的知识库中几乎不存在循环<sup>[2]</sup>, 因此本文假设构建的 IFPN 模型是一个非循环网.

假设 IFPR 集  $S$  中有  $n$  个命题,  $m$  条规则, 对应的 IFPN 模型有  $n$  个库所,  $m$  个变迁; 需要求解结果命题  $d_1, \dots, d_j$  的可信度, 对应的目标库所为  $p_i, \dots, p_j$ , 则基于 IFPN 的混合推理方法可由算法 1 和算法 2 构成, 具体如下:

#### 算法 1 构建推理子模型算法

**Input:** 输入输出转移矩阵  $I$ , 输出转移矩阵  $O$ , 关联库所向量  $AP_0$ , 关联变迁向量  $AT_0$ .

**Output:**  $AP_k, AT_k, k$ , 推理子模型 S-IFPN 及其定义.

预处理(判断 IFPN 模型中是否有回路): 在 IFPN 模型中, 若  $\exists p_i \in IRS(p_i) \cup RS(p_i), i=1, 2, \dots, n$ , 则模型中存在回路, 不能应用该推理算法, 退出.

**Step1** 令迭代次数  $k=1$ , 初始化所有输入,  $AP_{k-1} = (x_0^{k-1}, x_1^{k-1}, \dots, x_n^{k-1})^T = AP_0 = (x_0^0, x_1^0, \dots, x_n^0)^T$ , 其中向量  $AP_0$  中的元素  $x_i^0, \dots, x_j^0$  为 1, 其余元素为 0,  $AT_{k-1} = (y_0^{k-1}, y_1^{k-1}, \dots, y_m^{k-1})^T = AT_0 = (0, 0, \dots, 0)^T$ .

**Step2** 计算关联变迁向量  $AT_k = O * AP_{k-1}$

**Step3** 计算关联库所向量  $AP_k = I * AT_k + AP_{k-1}$

**Step4** 判断是否完全得到目标库所的关联库所和关联变迁. 如果  $AP_k = AP_{k-1}$  且  $AT_k = AT_{k-1}$ , 则已经完全得到目标库所的关联库所和关联变迁, 输出  $AP_k, AT_k, k$ , 并转到 Step5; 否则  $k = k + 1$ , 转到 Step2.

**Step5** 构建子模型

在原模型中删除向量  $AP_k$  和  $AT_k$  中 0 元素所对应的库所和变迁以及它们相关联的有向弧. 具体如下:

假设  $AP_k = (x_0^k, x_1^k, \dots, x_n^k)^T$  中的元素  $x_i^k = 0, i \in \{1, 2, \dots, n\}$  和  $AT_k = (y_0^k, y_1^k, \dots, y_m^k)^T$  中的元素  $y_j^k = 0, j \in \{1, 2, \dots, m\}$ , 则:

①删除原模型中的库所  $p_i$  以及  $p_i$  的转移弧  $\langle t_x, p_i \rangle$  和  $\langle p_i, t_x \rangle$ , 其中  $\langle t_x, p_i \rangle \in F, \langle p_i, t_x \rangle \in F$  且  $x = 1, 2, \dots, m$ ;

②删除原模型中的变迁  $t_j$  以及  $t_j$  的转移弧  $\langle p_y, t_j \rangle$  和  $\langle t_j, p_y \rangle$ , 其中  $\langle p_y, t_j \rangle \in F, \langle t_j, p_y \rangle \in F$  且  $y = 1, 2, \dots, n$ .

**Step6** 更新 S-IFPN  $(P', T', F'; I', O', \theta', Th', CF')$  中的元素, 其中

- ①删除库所集合  $P$  中的  $p_i$ , 得到子模型的库所集合  $P'$ ;
- ②删除变迁集合  $T$  中的  $t_j$ , 得到子模型的变迁集合  $T'$ ;
- ③  $F' = ((P' \times T') \cup (T' \times P')) \cap F$  为子模型的有向弧集合;
- ④删除  $I$  中的第  $i$  行和第  $j$  列, 得到子模型的  $I'$ ;
- ⑤删除  $O$  中的第  $j$  行和第  $i$  列, 得到子模型的  $O'$ ;
- ⑥删除  $\theta$  中的  $\theta_i$ , 得到子模型的库所 token 值向量  $\theta'$ ;
- ⑦删除  $Th$  中的  $\lambda_j$ , 得到子模型的变迁阈值向量  $Th'$ ;
- ⑧删除  $CF$  中的第  $j$  行和第  $j$  列, 得到子模型的  $CF'$ .

**Step7** 输出推理子模型 S-IFPN 及其定义.

#### 算法 2 求解目标库所的 token 值算法

**Input:** 输入和输出转移矩阵  $I'$  和  $O'$ , 变迁阈值  $Th'$ , 规则可信度  $CF'$ , 命题初始可信度  $\theta'^0$ .

**Output:** 库所的最终 token 值  $\theta'^l$ , 目标库所  $p_i, \dots, p_j$  的 token 值以及迭代次数  $l$ .

**Step1** 初始化所有输入, 令迭代次数  $l=1$ ,

$\theta'^{l-1} = \theta'^0 = (\theta_1^0, \theta_2^0, \dots, \theta_n^0)^T = (\langle \mu_1^0, \gamma_1^0 \rangle, \langle \mu_2^0, \gamma_2^0 \rangle, \dots, \langle \mu_n^0, \gamma_n^0 \rangle)^T$ , 若库所的初始 token 值未知, 用  $\langle 0, 1 \rangle$  表示, 初始等效输入

$\rho_{l-1} = \rho_0 = (\langle 0, 1 \rangle, \langle 0, 1 \rangle, \dots, \langle 0, 1 \rangle)^T$ .

**Step2** 计算各个变迁的等效输入, 即将各个变迁的所有输入库所的 token 值等效为单个输入库所的 token 值, 结果为

$$\rho_l = (\rho\theta_1^l, \rho\theta_2^l, \dots, \rho\theta_m^l)^T = (\langle \rho\mu_1^l, \rho\gamma_1^l \rangle, \langle \rho\mu_2^l, \rho\gamma_2^l \rangle, \dots, \langle \rho\mu_m^l, \rho\gamma_m^l \rangle)^T,$$

$$\text{其中 } \rho\mu_j^l = \min \left\{ x_i \mid x_i = \begin{cases} \mu_i^{l-1}, I'(p_i, t_j) = 1 \\ 1, I'(p_i, t_j) = 0 \end{cases} \right\},$$

$$\rho\gamma_j^l = \max \left\{ y_i \mid y_i = \begin{cases} \gamma_i^{l-1}, I'(p_i, t_j) = 1 \\ 0, I'(p_i, t_j) = 0 \end{cases} \right\}$$

$$\text{即 } \rho_l = (I'^T \otimes \theta'^{l-1}) \quad (5)$$

**Step3** 将各个的变迁等效输入与规则阈值进行比较, 保留可以使变迁触发的输入

$$\rho'_l = \rho_l Th' \quad (6)$$

**Step4** 计算变迁触发后输出库所的 token 值

①  $S_l = CF' \odot \rho'_l$ , 其中

$$S_l = (s_1^l, s_2^l, \dots, s_m^l)^T = (\langle s\mu_1^l, s\gamma_1^l \rangle, \langle s\mu_2^l, s\gamma_2^l \rangle, \dots, \langle s\mu_m^l, s\gamma_m^l \rangle)^T$$

$$\langle s\mu_j^l, s\gamma_j^l \rangle = \langle C\mu_j^l \times \rho\mu_j^l, C\gamma_j^l + \rho\gamma_j^l - C\gamma_j^l \times \rho\gamma_j^l \rangle$$

②  $Y_l = (y\theta_1^l, y\theta_2^l, \dots, y\theta_n^l)^T$

$$= (\langle y\mu_1^l, y\gamma_1^l \rangle, \langle y\mu_2^l, y\gamma_2^l \rangle, \dots, \langle y\mu_n^l, y\gamma_n^l \rangle)^T$$

$$\text{其中 } y\mu_i^l = \max \left\{ x_i \mid x_i = \begin{cases} s\mu_j^l, O'(p_i, t_j) = 1 \\ 0, O'(p_i, t_j) = 0 \end{cases} \right\},$$

$$y\gamma_i^l = \min \left\{ y_i \mid y_i = \begin{cases} s\gamma_j^l, O'(p_i, t_j) = 1 \\ 1, O'(p_i, t_j) = 0 \end{cases} \right\}$$

$$\text{即 } Y_l = (O' \otimes S_l) = O' \odot (CF' \odot \rho'_l) \quad (7)$$

**Step5** 计算所有库所的 token 值

$$\theta'^l = \theta'^{l-1} \oplus Y_l = (\theta_1^l, \theta_2^l, \dots, \theta_n^l)^T = (\langle \mu_1^l, \gamma_1^l \rangle, \langle \mu_2^l, \gamma_2^l \rangle, \dots, \langle \mu_n^l, \gamma_n^l \rangle)^T \quad (8)$$

**Step6** 判断推理是否结束

如果  $\theta'^l = \theta'^{l-1}$ , 推理结束, 输出最终 token 值  $\theta'^l$  和目标库所  $p_i, \dots, p_j$  的 token 值; 否则, 令  $l = l + 1$ , 转到 Step2.

## 3.2 混合推理方法分析

### 3.2.1 算法 1 的关键步骤分析

假设  $d_g$  是需要求解的命题, 对应的目标库所为  $p_g$ , 则在  $AT_0$  中,  $x_g^0 = 1$ . 当  $k=1$  时, 根据算法 1 可知:

(1)  $AT_1 = O \otimes AP_0 = (y_1^1, y_2^1, \dots, y_m^1)^T$ , 其中  $y_j^1 = \max_{1 \leq i \leq n} (O(t_j, p_g) \times x_g^0), j=1, 2, \dots, m$ , 当且仅当  $O(t_j, p_g) = 1$  和  $x_g^0 = 1$  同时成立时, 才有  $y_j^1 = 1$ , 由  $O(t_j, p_g) = 1$  可知  $t_j \in \cdot p_g$ , 这表示  $t_j$  是  $p_g$  的输入变迁, 即  $t_j$  是  $p_g$  的关联变迁.

$$(2) AP_1 = I \otimes AT_1 \oplus AP_0 = (x_1^1, x_2^1, \dots, x_n^1)^T$$

令  $Z_1 = I \otimes AT_1 = (z_1^1, z_2^1, \dots, z_n^1)^T$ , 其中  $z_i^1 = \max_{1 \leq j \leq m} (I(p_i, t_j) \times y_j^1), i=1, 2, \dots, n$ , 当且仅当  $I(p_i, t_j) = 1$  和  $y_j^1 = 1$  同时成立时, 有  $z_i^1 = 1$ ; 又  $AP_1 = Z_1 \oplus AP_0 = (x_1^1, x_2^1, \dots, x_n^1)^T$ , 由于  $x_g^0 = 1$ , 所以此时  $AP_1$  中的元素  $x_g^1 = 1, x_i^1 = 1$ , 由  $I(p_i, t_j) = 1$  可知  $p_i \in \cdot t_j$ , 这表示  $p_i$  是  $t_j$  的输入库所, 又由(1)知  $t_j \in \cdot p_g$ , 所以从  $p_g$  反向直接可达  $p_i$ , 即  $p_i$  是  $p_g$  的关联库所.

通过(1)和(2)可知  $t_j$  和  $p_i$  分别是  $p_g$  的关联变迁和关联库所. 同理, 当  $k=2, 3, \dots$  时, 按照步骤①和②计



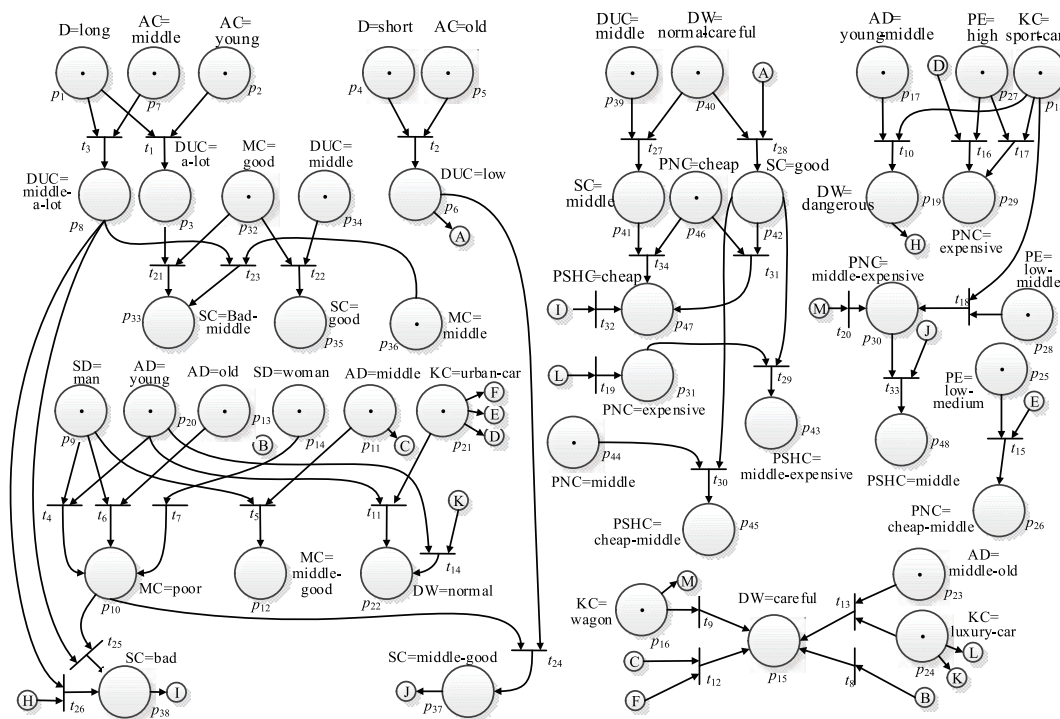


图6 规则库S1的IFPN模型

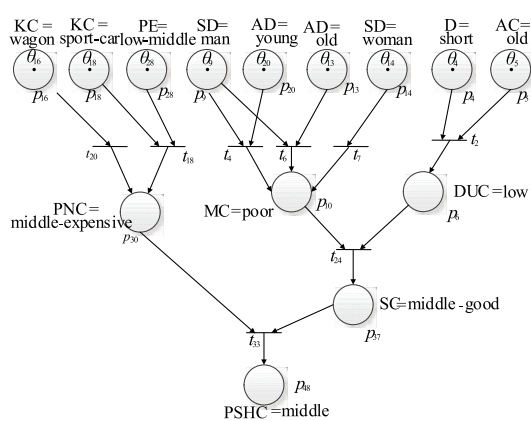


图7 推理子模型

假设

$$\theta^0 = (\theta_4^0, \theta_5^0, \theta_6^0, \theta_9^0, \theta_{10}^0, \theta_{13}^0, \theta_{14}^0, \theta_{16}^0, \theta_{18}^0, \theta_{20}^0, \theta_{28}^0, \theta_{30}^0, \theta_{37}^0, \theta_{48}^0)^T$$

$$= (\langle 0.9, 0.1 \rangle, \langle 0.8, 0.1 \rangle, \langle 0, 1 \rangle, \langle 1, 0 \rangle, \langle 0, 1 \rangle, \langle 0.6, 0.3 \rangle, \langle 0, 1 \rangle, \langle 0.8, 0.1 \rangle, \langle 0.7, 0.2 \rangle, \langle 0.9, 0.1 \rangle, \langle 0.6, 0.3 \rangle, \langle 0, 1 \rangle, \langle 0, 1 \rangle, \langle 0, 1 \rangle)^T$$

$$Th' = (\lambda_2, \lambda_4, \lambda_6, \lambda_7, \lambda_{18}, \lambda_{20}, \lambda_{24}, \lambda_{33})^T$$

$$= (\langle 0.6, 0.3 \rangle, \langle 0.5, 0.3 \rangle, \langle 0.5, 0.4 \rangle, \langle 0.7, 0.2 \rangle, \langle 0.4, 0.5 \rangle, \langle 0.6, 0.2 \rangle, \langle 0.5, 0.3 \rangle, \langle 0.5, 0.4 \rangle)^T$$

$$CF' = \text{diag}(CF_2, CF_4, CF_6, CF_7, CF_{18}, CF_{20}, CF_{24}, CF_{33})$$

$$= \text{diag}(\langle 1, 0 \rangle, \langle 1, 0 \rangle, \langle 1, 0 \rangle, \langle 1, 0 \rangle, \langle 1, 0 \rangle, \langle 1, 0 \rangle, \langle 1, 0 \rangle, \langle 1, 0 \rangle)$$

推理过程如下:

经过 4 次循环推理, 计算得

$$\theta^{t^4} = \theta^{t^3} = (\theta_4, \theta_5, \theta_6, \theta_9, \theta_{10}, \theta_{13}, \theta_{14}, \theta_{16}, \theta_{18}, \theta_{20}, \theta_{28}, \theta_{30}, \theta_{37}, \theta_{48})^T$$

$$= (\langle 0.9, 0.1 \rangle, \langle 0.8, 0.1 \rangle, \langle 0.8, 0.1 \rangle, \langle 1, 0 \rangle, \langle 0.9, 0.1 \rangle, \langle 0.6, 0.3 \rangle, \langle 0, 1 \rangle, \langle 0.8, 0.1 \rangle, \langle 0.7, 0.2 \rangle, \langle 0.9, 0.1 \rangle, \langle 0.6, 0.3 \rangle, \langle 0.8, 0.1 \rangle, \langle 0.8, 0.1 \rangle, \langle 0.8, 0.1 \rangle)^T$$

此时推理结束,  $P_{48}$  的 token 值为  $\theta_{48} = \langle 0.8, 0.1 \rangle$ , 即命题  $PSHC = middle$  的可信度为  $\langle 0.8, 0.1 \rangle$ , 它表示命题  $PSHC = middle$  为真的可能性为 0.8, 为假的可能性为 0.1, 不能确定真假的可能性为 0.1.

该推理结果与文献 [10] 的推理结果一致, 而且由于增加了非隶属度, 推理结果更细腻. 与文献 [10] 和 [11] 采用的基于图形的推理方法相比, 本文采用的基于矩阵的推理方法, 能充分利用 IFPN 的并行运算能力, 推理过程也更加简单.

### 4.2 例子 B

本节以文献 [2] 中的“涡轮机故障诊断专家系统”规则库  $S_2$  为例, 对上述推理方法进行验证. 规则库可参考文献 [2], 其对应的 IFPN 模型如图 8 所示.

假设用户想计算命题“压缩机处于湍流状态 ( $p_{18}$ )”的可信度并分析引起该故障的原因, 那么  $p_{18}$  即为目标库所, 推理过程如下:

(1) 首先运用算法 1 构建目标库所的关联子模型.

已知  $n = 28, m = 13$ , 输入转移矩阵  $I$  和  $O$ , 关联向量  $AP_0$  和  $AT_0$ . 构建子模型过程如下:

(a) 根据反向推理算法求出目标库所的关联库所

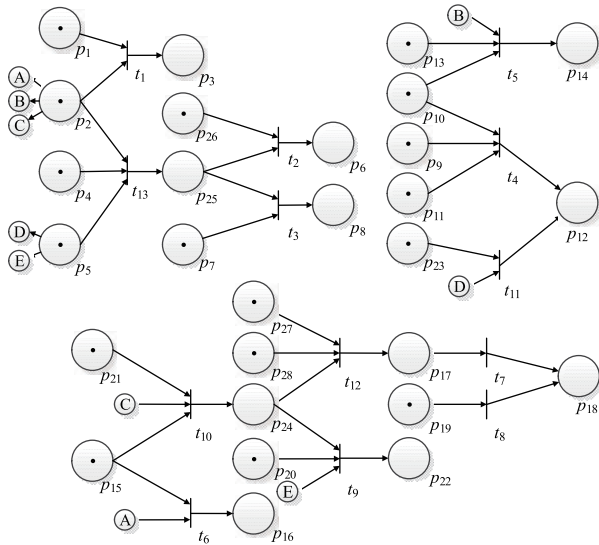


图8 涡轮机故障诊断系统的IFPN模型

向量和关联变迁向量。

经过4次循环推理,得  $AT_4 = AT_3 = (0,0,0,0,0,0,1,1,0,1,0,1,0)^T$ ,  $AP_4 = AP_3 = (0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,1,0,0,1,1)^T$ ,

由此可知  $IPS(p_{18}) = \{p_2, p_{15}, p_{17}, p_{19}, p_{21}, p_{24}, p_{27}, p_{28}\}$  为  $p_{18}$  的关联库所集,  $ITS(p_{18}) = \{t_7, t_8, t_{10}, t_{12}\}$  为  $p_{18}$  的关联变迁集。

(b) 构建子模型并更新子模型的定义。

简化后的推理子模型如图9所示。

子模型定义为 S-IFPN =  $(P', T', F'; I', O', \theta', Th', CF')$ , 其中  $P' = \{p_2, p_{15}, p_{17}, p_{18}, p_{19}, p_{21}, p_{24}, p_{27}, p_{28}\}$ ,  $T' = \{t_7, t_8, t_{10}, t_{12}\}$ ,

$$I' = \begin{matrix} p_2 \\ p_{15} \\ p_{17} \\ p_{18} \\ p_{19} \\ p_{21} \\ p_{24} \\ p_{27} \\ p_{28} \end{matrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad O' = \begin{matrix} p_{17} \\ p_{18} \\ p_{19} \\ p_{21} \\ p_{24} \\ p_{27} \\ p_{28} \end{matrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

(2) 接着根据算法2求解目标库所的 token 值。

假设

$$\theta^0 = (\theta_2, \theta_{15}, \theta_{17}, \theta_{18}, \theta_{19}, \theta_{21}, \theta_{24}, \theta_{27}, \theta_{28})^T = (\langle 0.6, 0.3 \rangle, \langle 0.7, 0.1 \rangle, \langle 0, 1 \rangle, \langle 0, 1 \rangle, \langle 0.8, 0.1 \rangle, \langle 0.7, 0.2 \rangle, \langle 0, 1 \rangle, \langle 0.4, 0.5 \rangle, \langle 0.6, 0.3 \rangle)^T,$$

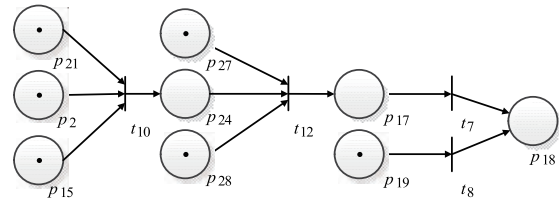


图9 简化后的推理子模型

$$Th' = (\lambda_7, \lambda_8, \lambda_{10}, \lambda_{12})^T = (\langle 0.6, 0.2 \rangle, \langle 0.6, 0.3 \rangle, \langle 0.5, 0.4 \rangle, \langle 0.7, 0.1 \rangle)^T,$$

$$CF' = \text{diag}(CF_7, CF_8, CF_{10}, CF_{12}) = \text{diag}(\langle 0.8, 0.1 \rangle, \langle 0.7, 0.2 \rangle, \langle 1, 0 \rangle, \langle 0.6, 0.3 \rangle).$$

推理过程如下:

经过2次循环推理,计算得

$$\rho'_2 = (\langle 0, 1 \rangle, \langle 0.8, 0.1 \rangle, \langle 0.6, 0.3 \rangle, \langle 0, 1 \rangle)^T, \theta'^2 = \theta'^1 = (\theta_2, \theta_{15}, \theta_{17}, \theta_{18}, \theta_{19}, \theta_{21}, \theta_{24}, \theta_{27}, \theta_{28})^T = (\langle 0.5, 0.3 \rangle, \langle 0.5, 0.4 \rangle, \langle 0, 1 \rangle, \langle 0.56, 0.28 \rangle, \langle 0.8, 0.1 \rangle, \langle 0.6, 0.3 \rangle, \langle 0.6, 0.3 \rangle, \langle 0.4, 0.5 \rangle, \langle 0.6, 0.3 \rangle)^T$$

此时推理结束. 目标库所的 token 值为  $\theta_{18} = \langle 0.56, 0.28 \rangle$ , 即“压缩机处于湍流状态( $p_{18}$ )”的可信度为  $\langle 0.56, 0.28 \rangle$ 。

下面对引起故障的原因进行分析:

根据算法2的 Step3 可知  $\rho'_i$  中的  $\langle 0, 1 \rangle$  元素表示对应的变迁未触发. 在本例中, 当推理结束时,  $\rho'_2 = (\langle 0, 1 \rangle, \langle 0.8, 0.1 \rangle, \langle 0.6, 0.3 \rangle, \langle 0, 1 \rangle)^T$ , 这说明变迁  $t_7$  和  $t_{12}$  在整个推理过程中未触发. 又根据定义8可知目标库所  $p_{18}$  有两个路径, 分别为变迁序列  $t_{10}, t_{12}, t_7$  和变迁  $t_8$ , 其中  $t_8$  为有效路径. 所以  $p_{18}$  只能通过  $t_8$  从源库所  $p_{19}$  中获取 token 值, 而不能通过变迁序列  $t_{10}, t_{12}, t_7$  从源库所  $p_2, p_{15}, p_{21}, p_{27}, p_{28}$  中获取 token 值, 这表示“压缩机处于湍流状态( $p_{18}$ )”这一故障产生的真正原因是“压缩机叶片损坏( $p_{19}$ )”。

如果运用文献[12,13,15,16]中的方法进行分析, 通过简化模型只能获知“组合单元效率太低( $p_2$ )”, “组合单元燃料消耗太高( $p_{21}$ )”, “涡轮机入口气体温度太高( $p_{15}$ )”, “压缩机的变换流量不太低( $p_{27}$ )”, “压缩机的增压率不太低( $p_{28}$ )”和“压缩机叶片损坏( $p_{19}$ )”都有可能 是导致“压缩机处于湍流状态( $p_{18}$ )”的原因, 而无法确定其产生的真正原因, 但是通过在模型中引入阈值并结合“路径”以及“有效路径”的定义后, 便可知该故障实际上是由“压缩机叶片损坏( $p_{19}$ )”引起的。

由此可以发现, 与文献[12,13,15,16]中的方法相比, 这些文献中的方法只能找出故障产生的可能原因, 而无法确定故障产生的真正原因, 而本文提出的方法在找出故障产生的可能原因后可以进一步排除无效的

故障原因,并找出故障产生的真正原因,从而提高了故障诊断的准确度,更利于故障的预防.

### 4.3 与现有方法的比较分析

通过实例验证可知,与现有方法相比较,本文提出的方法具有以下优点:

(1) 克服了 FPN 隶属度单一的缺陷.

通过将直觉模糊理论与 Petri 网理论相结合,克服了 FPN 隶属度单一的缺陷,并扩展了 FPN.

(2) 在求解单个或部分问题时,该方法能有效简化推理模型,提高正向推理效率.

假设原 IFPN 模型库所数为  $n$ ,变迁数为  $m$ ,简化后的子模型库所数为  $n'$ ,变迁数为  $m'$ ,根据算法 1 可知,子模型的库所和变迁数通常都比原模型少,所以与原模型相比,运用简化后的子模型进行推理能有效简化正向推理过程,从而提高效率.

(3) 推理过程简单高效.

与文献[10,11]提出的基于图形的推理方法相比,本文提出的混合推理方法通过矩阵运算实现推理,运算过程虽没有文献[10,11]中的方法直观,但能充分利用 Petri 网的并行运算能力和 IFPN 的推理能力,比文献[10,11]中的方法的推理过程更简单、更高效.

(4) 能排除无效原因.

与文献[12,13,15,16]中的方法相比,本文提出的方法在得出故障产生的可能原因后,通过引入阈值并结合“路径”以及“有效路径”等定义,可以进一步排除无效故障并找出真正的故障原因.

(5) 方法具有通用性.

在已知原 IFPN 推理模型和目标库所  $p_g$  的前提下,算法 1 仅使用原模型的输入转移矩阵  $I$  和输出转移矩阵  $O$  参与运算,在得出目标库所的关联库所集  $IPS(p_g)$  和关联变迁集  $ITS(p_g)$  后,只需结合原 IFPN 模型及其定义删除多余的库所、变迁和有向弧等元素即可得到子模型 S-IFPN 及其定义,所以该方法的算法 1 也可用于其他类型的 IFPN 和 FPN,具有一定的通用性.

## 5 结论

本文提出的基于 IFPN 的混合推理方法综合运用了反向推理和正向推理.理论分析和实例验证表明,当在推理过程中仅需针对特定的问题进行分析时,该方法不仅可以有效简化推理模型,提高推理效率,而且在推理结束后可以对问题产生的原因进行分析,并能从潜在的原因中找出问题产生的真正原因.该方法在复杂 FPN 和 IFPN 推理模型的简化、故障诊断和异常检测等领域有着良好的应用前景.

### 参考文献

[1] Shyi-Ming Chen, Jyh-Sheng Ke, Jin-Fu Chang. Knowledge

representation using fuzzy Petri nets [J]. IEEE Transaction on Knowledge and Data Engineering, 1990, 2 (3): 311 - 319.

[2] Meimei Gao, Mengchu Zhou, Xiaoguang Huang, Zhiming Wu. Fuzzy reasoning Petri nets [J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2003, 33(3): 314 - 324.

[3] 汪洋,林闯,曲扬,等.含有否定命题逻辑推理的一致性模糊 Petri 网模型[J].电子学报,2006,34(11):1955 - 1960.

Wang Yang, Ling Chuang, Qu Yang, et al. Consistent fuzzy Petri nets model for logic programs with negation [J]. Acta Electronica Sinica, 2006, 34(11): 1955 - 1960. (in Chinese)

[4] Xiaoou Li, Wen Yu, Felipe Lara-Rosano. Dynamic knowledge inference and learning under adaptive fuzzy Petri net framework [J]. IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, 2000, 30 (4): 442 - 450.

[5] Hu-Chen Liu, Long Liu, Q, et al. Knowledge acquisition and representation using fuzzy evidential reasoning and dynamic adaptive fuzzy Petri nets [J]. IEEE Transactions on Cybernetics. 2013, 43(3): 1059 - 1072.

[6] Feng Zhou, Roger J. Jiao, Qianli Xu, et al. User experience modeling and simulation for product ecosystem design based on fuzzy reasoning Petri nets [J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans. 2012, 1(42): 201 - 212.

[7] Xiao-yong Shen, Ying-jie Lei, Cheng-hai Li. Intuitionistic fuzzy Petri nets model and reasoning algorithm [A]. 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery [C]. Tianjin, IEEE, 2009. 119 - 122

[8] Hu-Chen Liu, Jian-Xin You, Xiao-Yue You, et al. Fuzzy Petri nets using intuitionistic fuzzy sets and ordered weighted averaging operators [J]. IEEE Transactions on Cybernetics, DOI:10.1109/TCYB.2015.2455343.

[9] 孟飞翔,雷英杰,余晓东,等.基于直觉模糊 Petri 网的知识表示和推理[J].电子学报,2016,44(1):77 - 86  
MENG Fei-xiang, LEI Ying-jie, YU Xiao-dong, et al. Knowledge representation and reasoning using intuitionistic fuzzy Petri nets [J]. Acta Electronica Sinica, 2016, 44(1): 77 - 86. (in Chinese)

[10] Shyi-Ming Chen. Fuzzy backward reasoning using fuzzy Petri nets [J]. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 2000, 30(6): 846 - 856.

[11] H Scarpelli, F Gomide, R R Yager. A reasoning algorithm for high-level fuzzy Petri nets [J]. IEEE Transactions on Fuzzy Systems. 1996, 4(3): 282 - 294.

[12] 鲍培明.模糊 Petri 网模型的反向推理算法[J].南京师

- 范大学学报(工程技术版),2003,3(3):21-25
- Bao Peiming. Reverse reasoning algorithm based on the fuzzy petri net model [J]. Journal of Nanjing Normal University (Engineering and Technology), 2003, 3(3): 21-25. (in Chinese)
- [13] Jie Yuan, Hai-bo Shi, Chang Liu, et al. Backward concurrent reasoning based on fuzzy petri nets [A]. 2008 IEEE International Conference on Fuzzy Systems [C]. Hong Kong: IEEE, 2008. 832-837
- [14] Yan Ye, Zhibin Jiang, Xiaodi Diao, et al. Extended event-condition-action rules and fuzzy Petri nets based exception handling for workflow management [J]. Expert Systems with Applications, 2011, 38: 10847-10861
- [15] Hesuan Hu, Zhiwu Li, Abdulrahman Al-Ahmari. Reversed fuzzy Petri nets and their application for fault diagnosis [J]. Computers & Industrial Engineering, 2011, 60: 505-510.
- [16] Hu-Chen Liu, Qing-Lin, Ming-LunRen. Fault diagnosis and cause analysis using fuzzy evidential reasoning approach and dynamic adaptive fuzzy Petri nets [J]. Computers & Industrial Engineering, 2013, 66: 899-908.

### 作者简介



**孟飞翔** 男,1986 年出生,河南信阳人,现为空军工程大学计算机应用专业博士研究生,主要研究方向为智能信息处理。  
E-mail: ttimo@163.com



**雷英杰** 男,1956 年出生,陕西华阴人,现为空军工程大学教授,博士生导师,主要研究方向为智能信息处理与智能决策。  
E-mail: leiyjie@163.com