Expert Systems with Applications 42 (2015) 3980–3990

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A decomposition algorithm of fuzzy Petri net using an index function and incidence matrix

Kai-Qing Zhou [a,*], Azlan Mohd Zain [a], Li-Ping Mo [b]

[a] Soft Computing Research Group, Faculty of Computing, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
[b] College of Information Science and Engineering, Jishou University, 416000 Jishou, Hunan, PR China

ARTICLE INFO

Article history:
Available online 7 January 2015

Keywords:
Fuzzy Petri net
Decomposition algorithm
Index function
Incidence matrix
Inference path

ABSTRACT

As with Petri nets (PNs), the state space explosion has limited further studies of fuzzy Petri net (FPN), and with the rising scale of FPN, the algorithm complexity for related applications using FPN has also rapidly increased. To overcome this challenge, we propose a decomposition algorithm that includes a backwards search stage and forward strategy for further decomposition, one that divides a large-scale FPN model into a set of sub-FPN models using both a presented index function and incidence matrix. In the backward phase, according to different output places, various completed inference paths are recognized automatically. An additional decomposition operation is then executed if the "OR" rule exists for each inference path. After analysing the proposed algorithm to confirm its rigor, a proven theorem is presented that calculates the number of inference paths in any given FPN model. A case study is used to illustrate the feasibility and robust advantages of the proposed decomposition algorithm.

1. Introduction

For the past decades, the Petri net (PN) has been successfully applied in various industrial fields (Murata, 1989; Urawski & Zhou, 1994). However, along with increased complexity of the sophisticated system, the homologous scale of the PN model has significantly grown as well. This shortfall is called 'state space explosion,' which has served to limit exhaustive studies and further implementation of the PN model. To overcome this challenge, numerous simplifications (decomposition algorithms) for the PN model have been based on and proposed for different application environments. For example, Berthelot and Terrat (1982) presented a technique to divide a complex PN model into a group of small-scale, easy-to-analyse sub-models. Murata and Koh (1980) proposed several reduction methods for T-graph and discussed the consistence of dynamic properties between T-graph and sub-models. Lee-Kwang, Favrel, and Baptiste (1987) employed six reduction rules to control the scale of the PN model. Li, Zhou, and Dai (2012) introduced an algebraic reduction operation for the general transformation of nets and applied the approach to model and analyse a mail sorting system. Shen, Chung, Chen, and Guo (2013) also designed an effective reduction approach for Petri net using matching theory.

Fuzzy Petri net (FPN), which is one type of high level Petri net (HLPN), is proposed in this research to model, analyse, and implement inference for knowledge-based systems (KBSs) or systems with uncertainty (Looney, 1988). Up to now, FPN has been applied widely for several industrial areas, such as traffic engineering (Asthana, Ahuja, & Darbari, 2011; Barzegar, Davoudpour, Meybodi, Sadeghian, & Tirandazian, 2011; Cheng & Yang, 2009), abnormality monitoring (Liu, Li, & Zhou, 2011), workflow management (Gong & Wang, 2012; Ye, Jiang, Diao, & Du, 2011), robotics engineering (Wai & Liu, 2009; Wai, Liu, & Lin, 2010), etc.

However, the implementation of inference using the FPN model became more difficult due to the increasing of the scale of that model (state space explosion issue). These difficulties of FPNs explored in this research are as follows:

1. With the growing of the scale of the FPN model, the number of requisite parameters is also increased. In recent literature, the values of parameters are determined based on the experience of the related experts. This also indicates that the accuracy of the reasoning result is hard to control because of the increasing number of parameters of the FPN model.
2. The existing reasoning algorithms can be separated into two main mechanisms, namely reasoning by using spurting tree and reasoning by algebraic analysis. However, dimensions of the spurting tree or related matrices/vectors also depend on the scale of the FPN model. Therefore, the dimensions of these

* Corresponding author.
E-mail addresses: jsu_computer@163.com (K.-Q. Zhou), azlanmz@gmail.com (A.M. Zain), zmx89@163.com (L.-P. Mo).

http://dx.doi.org/10.1016/j.eswa.2014.12.048
0957-4174/

matrices and vectors are all increased with the growing of the scale of the FPN model.

Focusing on these two challenges, a possible method to simplify the inference process is to divide the large-size FPN model into a series of completed reasoning paths (sub-FPN models) for each of the output places. As a backward compatible extension of PN, FPN preserves useful PN properties while extending its descriptive ability to fuzzy data. Hence, existing outcomes from the PN model can be utilized to construct a related decomposition algorithm for the FPN model. However, the biggest difference between FPN and other PNs is that there are various inference paths in the FPN model. Hence, their inner-reasoning relationship will be destroyed if the entire FPN model is decomposed using the existing algorithms mentioned above. Focusing on this characteristic, a novel biphasic decomposition algorithm for the FPN model is presented in this paper after a systematic analysis of the existing decomposition algorithm. The proposed decomposition algorithm divides an FPN into a set of completed, separated and irrelevant sub-FPN models (inference paths). The main contributions of this effort are summarized as follows.

1. A two-purpose index function is proposed to record the pre-set placements in the FPN model and judge whether the "OR" rule exists;
2. A biphasic decomposition algorithm is presented that divides the FPN model using the proposed index function and incidence matrix;
3. A proven theorem is presented to calculate the number of inference paths in any given FPN model.

The remaining sections of this paper are organized as follows. Section 2 discusses concepts related to FPN and the fuzzy production rule (FPR). Section 3 describes the implementation process of the proposed algorithm. Section 4 reviews details of our algorithm and provides the theorem just cited. In Section 5, a case study is presented that illustrates the algorithm's implementation. Section 6 presents conclusions and recommendations for future work.

2. Fuzzy Petri net and fuzzy production rule

In this section, FPN and FPR formalisms are described and the corresponding FPN models for each FPR type are generated, after which concepts related to the proposed algorithm are introduced.

2.1. Fuzzy Petri net

An 8-tuple formalism of FPN is selected for implementation in the proposed algorithm.

Definition 1 (*Fuzzy Petri net (FPN)*). FPN is represented by an 8-tuple $\sum = \{P, T, M, I, O, W, \mu, CF\}$, where

1. $P = \{p_1, p_2, \cdots, p_n\}$ is a finite set of places. In this formalism, P is classified into three sets, which are P_{in}/ P_{middle} / P_{out}.

P_{in}	set of input places;
P_{out}	set of output places;
P_{middle}	set of places except input and output places.

Moreover, $P_{in} \cap P_{middle} \cap P_{out} = \varnothing$ and $P_{in} \cup P_{middle} \cup P_{out} \neq \varnothing$.

2. $T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions.
3. $M = (m_1, m_2, \ldots, m_n)^T$ is a vector of fuzzy marking. $m_i \in [0, 1]$ is the truth degree of p_i $(i = 1, 2, \ldots, n)$. The initial truth degree vector is denoted by M_0.
4. $I : P \times T \to \{0, 1\}$ is the $n \times m$ input matrix. $I(p_i, t_j)$ records whether a directed arc from p_i to t_j exists $[(i = 1, 2, \ldots, n; j = 1, 2, \ldots, m)]$, where
$$I(p_i, t_j) = \begin{cases} 1 & \text{if there is a directed arc from } \mathrm{p_i} \text{ to } \mathrm{t_j} ; \\ 0 & \text{if there is not a directed arc from } \mathrm{p_i} \text{ to } \mathrm{t_j}. \end{cases}$$
5. $O : P \times T \to \{0, 1\}$ is the $n \times m$ output matrix. $O(p_i, t_j)$ records whether a directed arc from t_j to p_i exists $[(i = 1, 2, \ldots, n; j = 1, 2, \ldots, m)]$, where
$$O(p_i, t_j) = \begin{cases} 1 & \text{if there is a directed arc from } \mathrm{t_j} \text{ to } \mathrm{p_i}; \\ 0 & \text{if there is not a directed arc from } \mathrm{t_j} \text{ to } \mathrm{p_i}. \end{cases}$$
6. $w(i, j)$ is the weight of the arc from p_i to t_j .
7. $\mu : \mu \to (0, 1]$, μ_j is the threshold of t_j ;
8. CF is the belief strength, where $CF_{is} \in [0, 1]$ is the support strength of the arc from t_j to p_s $(s = 1, 2, \ldots, n;\ j = 1, 2, \ldots, m)$.

2.2. Fuzzy production rule (FPR)

FPR is a type of production rule that describes the interior relationship between pre-positions and conclusions with fuzzy parameters (Bandler, 1985; Yeung & Tsang, 1997; Tsang & Yeung, 1997).

Definition 2 (*Fuzzy production rule*). The general FPR formalism is described as follows:

if $D(\lambda)$ *then* Q (CF, μ, w), *where*

1. D is a finite set of preconditions, $D = \{D_1, D_2, \ldots, D_n\}$;
2. Q is a finite set of conclusions, $Q = \{Q_1, Q_2, \ldots, Q_n\}$;
3. λ is the truth degree of each precondition, $\lambda \in [0, 1]$;
4. CF is the belief strength of this rule, where $CF \in [0, 1]$ is the credibility after implementation of the rule;
5. μ is the threshold of the rule, $\mu \in [0, 1]$;
6. w is the weight of each precondition, $w \in [0, 1]$.

FPRs can be classified into three types: Simple, AND, or OR.

Type 1: Simple rule

if $D(\lambda)$ *then* Q $(CF, \mu, w = 1)$

If $\lambda > \mu$ exists, then the rule can be fired. The corresponding FPN model of a simple rule is generated as shown in Fig. 1(a), and the result of a fired rule after firing is illustrated in Fig. 1(b).

Type 2: 'AND' rule

if $D_1(\lambda_1)$ *and* $D_2(\lambda_2)$ *and* $\cdots$ *and* $D_n(\lambda_n)$ *then* Q $(CF, \mu, \sum w_i = 1)$

If $\sum w_i \lambda_i \geqslant \mu$ exists, then the rule can be fired. The corresponding FPN model of the "AND" rule is generated as shown in Fig. 2(a), and the result of a fired rule after firing is illustrated in Fig. 2(b).

Type 3: 'OR' rule

if $D_1(\lambda_1)$ *or* $D_2(\lambda_2)$ *or* $\cdots$ *or* $D_n(\lambda_n)$ *then* Q $(CF_i, \mu_i, w_i = 1)$

If $w_i \lambda_i > \mu_i$ exists, then the rule can be fired. The corresponding FPN model of the "OR" rule is generated as shown in Fig. 3(a), and the result of a fired rule after firing is illustrated in Fig. 3(b).

2.3. Other related notions in the proposed algorithm

Definition 3 (*Pre-set and post-set*). For FPN $\Sigma = \{P, T, M, I, O, W, \mu, CF\}$, ${}^\bullet x = \{y | (y, x) \in F\}$ is the pre-set or input set of x and $x^\bullet = \{y | (x, y) \in F\}$ is the post-set or output set of x, $x, x, y \in p \cup T$. F is a flow relationship of the FPN model.