Divide Large Scale Fuzzy Petri Net into Equivalent Sub-FPNs using Reachability Tree

Bi-Yu Liu¹, Kai-Qing Zhou¹*, Sharifah Zarith Rahmah Syed Ahmad

¹. College of Information Science and Engineering, Jishou University, Jishou, 416000, Hunan, China
². School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Skudai, 80310, Johor, Malaysia
Email : kqzhou@jsu.edu.cn

Abstract. As real systems become larger and complicated, the number of nodes of fuzzy Petri net (FPN) also increases rapidly, it further indicated that the algorithm complexity of related application using FPN grows up sharply. To solve this state-explosion issue, a decomposition algorithm using reachability tree is proposed to divide a large-scale FPN model into a series of sub-FPNs in this paper. Comparison experiments show that all obtained sub-FPNs do not destroy internal reasoning paths of the original large-scale FPN and represent a complete reasoning path from input place(s) to the output place.

1. Introduction

Fuzzy Petri net (FPN) is one kind of high-level Petri nets (HLPNs) based on the backward extension principle [1]. On the basis of the graphical description capability of traditional PN and the mathematical analysis mechanism of system, FPN can accurately depict the uncertainty and is widely used in the modeling, analyzing, and reasoning of knowledge-based systems (KBSs) or uncertainty systems in the engineering field, such as fault diagnosis for integrated manufacturing systems, traffic engineering control, workflow management, and power system diagnosis [2]. As with the PNs, the further study of FPN applications is also hindered by the "state space explosion". With the increase in the scale of FPNs, the state space explosion increases the algorithm complexity of the analysis and reasoning algorithms based on FPNs. The main issues can be attributed to two aspects.

First, as the scale of FPNs expands, the number of parameters required in the reasoning process increases. The acquisition of various parameter values of FPN models depends on relevant expertise. Therefore, the more complicated the FPN models are, the more difficult it is to control the accuracy of reasoning results [3].

Second, the existing FPN-based reasoning algorithms can be divided into two categories, namely, the use of reachability trees and the reasoning based on algebraic analysis [4]. The scale of reachability trees or the associated matrix/vectors is also increased as the scale of the FPNs expands.

A feasible FPN decomposition idea to solve the two problems is to divide large-scale FPN models into a series of complete FPN sub models (i.e., reasoning path sets) based on each output place (i.e., the final reasoning results). FPN, as a HLPN formalism based on the backward extension principle, can also describe fuzzy information in addition to the inherent characteristics of PN. Therefore, pertinent decomposition algorithms of FPNs can be constructed from the existing ones of PNs. However, various reasoning paths exist in FPNs. If an FPN model is decomposed directly using an
existing PN decomposition algorithm, its internal reasoning relation will be destroyed. In view of this situation, Zhou et al. [5] proposed an FPN decomposition algorithm based on index function and incidence matrix and presented a computational method for calculating the number of final subnets in a large-scale FPN model. This method achieved the equivalent decomposition of FPNs based on the algebraic analysis capability of FPN, but its process was complicated and required large amounts of computation. This manuscript proposes a decomposition algorithm using topological sort among nodes of corresponding reachability tree of FPN. The decomposition algorithm aims at dividing FPN into a set of complete, separate, and uncorrelated sub-FPN models through the topological sort of the and-or reachability tree nodes that correspond to FPNs without destroying the inner-inference-path.

The organization of the rest sections is as follows. Section 2 discusses the concepts of FPN, FPR and reachability tree. Section 3 illustrates the proposed decomposition algorithm using topological sorts among nodes of reachability tree nodes. A case study is used to prove the validity and feasibility of the presented algorithm. Sections 5 concludes and gives a brief conclusion.

2. Fuzzy Production Rule (FPR), FPN, and Reachability Tree

This section introduces the related concepts involved in the proposed algorithm, including FPR, FPN, and reachability tree.

FPR regular expression is a commonly used knowledge representation method, which is mainly used to describe the relevant uncertainties in expert systems. General FPRs are formalized as

\[
\text{if } D(\lambda) \text{ then } Q(CF, \mu, w)
\]

Where, \( D \) is a limited set of preconditions, \( D = \{D_1, D_2, \ldots, D_n\} \); \( Q \) is a limited set of conclusions, \( Q = \{Q_1, Q_2, \ldots, Q_m\} \); \( \lambda \) is the true extent of each precondition, \( \lambda \in [0,1] \); \( CF \) is the credibility of the rule; \( \mu \in [0,1] \) is the credibility of the conclusion obtained after the rule is executed; \( \mu \) is the threshold of the rule, \( \mu \in (0,1] \); and \( w \) is the weight of each precondition, \( w \in (0,1] \).

Where, \( P = \{p_1, p_2, \ldots, p_n\} \) represents a finite set of places, where \( n \) represents the number of places in the rule; \( T = \{t_1, t_2, \ldots, t_m\} \) represents a finite set of transitions, where \( m \) represents the number of transitions in the rule; \( I(O) \) is the input (output) function, i.e., the mapping relation between places and transitions; \( M = (m_1, m_2, \ldots, m_n)^T \) indicates the identity of places; \( w_i \) indicates the weight of places \( p_i \), i.e., the support degree for the rule establishment by preconditions \( p_i \); \( CF \) indicates the credibility, i.e., the true extent of the conclusion after transitions \( t_j \) fired; and \( \mu : \mu \rightarrow (0,1], \mu_i \) is the threshold of transitions \( t_j \).

Reachability tree is a method used for problem reduction, which expresses complex problems with several simple problems. If the original problem can only be solved when solutions of multiple simple problems are simultaneously established, the relationship among these simple problems is and relation; if the original problem can be solved when multiple simple problems are established, the relationship among these simple problems is or relation; if the original problem can be solved when any one simple problem is established, the relationship is called simple relation. The problem that can no longer be resolved is called the original problem.

3. Algorithm Thoughts

The purpose of FPN model decomposition is to identify each of the simplest reasoning paths (i.e., FPN subnet sets) one by one without destroying the intrinsic reasoning paths of FPN models. An FPN model contains two types of nodes (places and transitions). Therefore, this study obtains the relationship among the places in the FPN through the topological sort of its equivalent reachability tree nodes. If a certain node exists and has been acquired in the reachability tree node sequence, its
corresponding parent nodes must also exist in the sequence. Therefore, if a non-leaf node of the and–or tree is selected into a certain path, one or all of the nodes that correspond to the non-leaf node must be stored in the path according to the and–or relation among the nodes.

The preceding analysis indicates that each FPN subnet can be obtained by traversing root nodes in the topological sort of the and–or tree nodes. As mentioned above, the decomposition algorithm proposed in this study includes two phases.

First, the topological sort of reachability tree nodes is obtained according to equivalent and–or tree models with decomposed FPNs. Second, each root node is traversed according to the and–or relation among the nodes to achieve the equivalent decomposition of FPN. The complete flowchart of the decomposition algorithm is shown in figure 1.

---

**Figure 1.** Overall flowchart of the FPN decomposition algorithm

4. Experiment and Analysis

The purpose of decomposition algorithm is to identify each simplest reasoning path one by one without destroying the intrinsic reasoning paths of FPN models. An FPN model contains two types of nodes (places and transitions). Therefore, this study obtains the relationship among the places in the FPN through the topological sort of its equivalent reachability tree nodes. The FPN model of
Literature [5] is shown in figure 2 (Relevant parameter values are omitted.). The corresponding and–or tree model according to the FPN model shown in figure 2 is depicted in figure 3.

The proposed decomposition algorithm is performed on the reachability tree shown in figure 3 to obtain tables 1 and 2. Table 1 records the relevant information of each root node, which includes the corresponding child nodes of the root nodes, the serial numbers of the child nodes, and the relationship among the child nodes (In the table, 1 represents ‘and’ relation, and 0 represents ‘or’/simple relation.). Table 2 presents the node information of all decomposed subnets obtained from topological sort.

| Serial number of root node | Quantity of child nodes | Serial number(s) of child nodes | Relationship among child nodes |
|----------------------------|-------------------------|---------------------------------|-------------------------------|
| C3                         | 3                       | R2, R32, R33                   | 1                             |
| C7                         | 1                       | C3                              | 0                             |
| C10                        | 1                       | C7                              | 0                             |
| C11                        | 1                       | C7                              | 0                             |
| C13                        | 1                       | C11                             | 0                             |
| C12                        | 2                       | C8, C10                         | 1                             |
| C8                         | 2                       | C4, C5                          | 0                             |
| C5                         | 1                       | C2, C3                          | 0                             |
| C2                         | 1                       | R2                              | 0                             |
| C4                         | 2                       | C1, R42                         | 1                             |
| C1                         | 1                       | R1                              | 0                             |
Table 2. Node information of each subnet obtained after decomposition

| Subnet serial number | Subnet 1 | Subnet 2 | Subnet 3 | Subnet 4 |
|----------------------|----------|----------|----------|----------|
| C13                  | 1        | 0        | 0        | 0        |
| C12                  | 0        | 1        | 1        | 1        |
| C11                  | 1        | 0        | 0        | 0        |
| C10                  | 0        | 1        | 1        | 1        |
| C8                   | 0        | 1        | 1        | 1        |
| C7                   | 1        | 1        | 1        | 1        |
| C5                   | 0        | 1        | 0        | 1        |
| C4                   | 0        | 0        | 1        | 0        |
| C2                   | 0        | 1        | 0        | 0        |
| C3                   | 1        | 1        | 1        | 1        |
| R42                  | 0        | 0        | 1        | 0        |
| C1                   | 0        | 0        | 1        | 0        |
| R33                  | 1        | 1        | 1        | 1        |
| R32                  | 1        | 1        | 1        | 1        |
| R2                   | 1        | 1        | 1        | 1        |
| R1                   | 0        | 0        | 1        | 0        |

From tables 1 and 2, four Fuzzy Petri subnets can be obtained from the FPN model in figure 2 through the proposed decomposition algorithm, as shown in figure 4 (a)–(d).

Figure 4 illustrates that the FPN model in figure 6 is equivalently decomposed into four sub-FPNs, which is consistent with the experimental results of literature [5]. Therefore, the decomposition algorithm proposed in this study can effectively and accurately decompose the FPN model into subnets using the topological sort of reachability tree nodes without destroying the integrity of the reasoning paths to alleviate the problem of "space state explosion" in FPNs to some extent.
5. Conclusion
In this research, a new decomposition algorithm is proposed to divide an FPN model into a series of completed sub-FPN models using a proposed index function and incidence matrix. The index function is used to judge whether the “OR” rule exists in an inference path. This two-phase decomposition algorithm can facilitate analysing and obtaining the existing inference paths of an FPN model. The proof theorem demonstrated how to calculate the number of inference paths of an FPN model.

Acknowledgement
This work was supported by the Research Foundation of Education Bureau of Hunan Province (Nos 18B317 and 19A414), Research-based Study and Innovative Experimental Project for College Students in Hunan Province (No.20180592), National Natural Science Foundation of Hunan Province (No.2019JJ40234) and Special Project on Language Application of Hunan Provincial Language Committee (No. XYJ2019GB09).

References
[1] Zhou K Q, Zain A M, Mo L P 2015 Dynamic properties of fuzzy Petri net model and related analysis J Cent South Univ 22 pp. 4717-23.
[2] Liu H C, You J X, Li Z W and Tian G 2017 Fuzzy Petri nets for knowledge representation and reasoning: A literature review J Eng Appl Artificial Intel 60 pp 45-56.
[3] Zhou K Q, Mo L P, Jin J and Zain A M 2019 An equivalent generating algorithm to model fuzzy Petri net for knowledge-based system J Intel Manuf 30 pp 1831-42.
[4] Zhou K Q, Mo L P, Ding L and Gui W H 2018 An Automatic Algorithm to Generate a Reachability Tree for Large-Scale Fuzzy Petri Net by And/Or Graph Symmetry 10 p 454.
[5] Zhou K Q, Zain A M and Mo L P 2015 A decomposition algorithm of fuzzy Petri net using an index function and incidence matrix 2 J Expert Syst Appl 42 pp 3980-90.