HIGH-LEVEL FUZZY PETRI NETS AS A BASIS FOR MANAGING SYMBOLIC AND NUMERICAL INFORMATION

KEVIN F.R. LIU  
Department of Environmental Engineering  
Da-Yeh University  
Changhua, Taiwan 51542

JONATHAN LEE  
Department of Computer Science and Information Engineering  
National Central University  
Chungli, Taiwan 32054

WEILING CHIANG  
Department of Civil Engineering  
National Central University  
Chungli, Taiwan 32054

Received December 1999  
Revised July 2000  
Accepted July 2000

The focus of this paper is on an attempt towards a unified formalism to manage both symbolic and numerical information based on high-level fuzzy Petri nets (HLFPN). Fuzzy functions, fuzzy reasoning, and fuzzy neural networks are integrated in HLFPN. In HLFPN model, a fuzzy place carries information to describe the fuzzy variable and the fuzzy set of a fuzzy condition. An arc is labeled with a fuzzy weight to represent the strength of connection between places and transitions. A fuzzy set and a fuzzy truth-value are attached to an uncertain fuzzy token to model imprecision and uncertainty. We have identified six types of uncertain transition: calculation transitions to compute functions with uncertain fuzzy inputs; inference transitions to perform fuzzy reasoning; neuron transitions to execute computations in neural networks; duplication transitions to duplicate an uncertain fuzzy token to several tokens carrying the same fuzzy sets and fuzzy truth values; aggregation transitions to combine several uncertain fuzzy tokens with the same fuzzy variable; and aggregation-duplication transitions to amalgamate aggregation transitions and duplication transitions. To guide the computation inside the HLFPN, an algorithm is developed and an example is used to illustrate the proposed approach.

Keywords: Fuzzy reasoning, fuzzy neural networks, high-level fuzzy Petri nets.

1. Introduction

Usually, a complex system in the real world will be required to manage both symbolic and numerical information at the same time. However, there is no single modeling formalism that can well serve the request. Therefore, we usually break
down a complex system into several subsystems based on their characteristics. For example, one part of a system can be described precisely via a mathematical function if this subsystem is well studied; whereas, another part of the system can be presented by means of rules if the heuristic knowledge has a critical role in this subsystem; furthermore, some attempts have been made on constructing the relationships between the inputs and the outputs of a subsystem through artificial neural networks if the arithmetic and the heuristic paradigms are inappropriate. To model a complicated system, a synthesis of diverse modeling tools is in need. Furthermore, a lack of complete understanding about the mechanisms of the subsystems or precise measurement makes the subsystems or their input/output invaded by uncertainty and imprecision.

We have examined a number of related literatures that model fuzzy reasoning through the use of fuzzy Petri nets. Looney’s approach did not allow partial matching and changed the firing rule. In Chen et al.’s approach, only exact matching was allowed. Bugarin et al.’s approach was not appropriate for large systems since the arrangement of the linking transitions in a net and applied algorithm depended on the initial markings. Konar et al.’s approach adopted Looney’s modifications on firing rule and made their algorithm inconsistent with the execution of the Petri net. Scarpelli et al. proposed an algorithm to extract the subnet from the entire net, then the subnet with concurrency could not be executed as a Petri net since only one path was shown. We have proposed our version of fuzzy Petri nets for modeling fuzzy rule-based reasoning that brings together the possibilistic entailment and the fuzzy reasoning to handle uncertain and imprecise information. A reasoning algorithm based on fuzzy Petri nets was also presented to improve the efficiency of fuzzy reasoning. The reasoning algorithm was consistent with not only the rule-based reasoning but also the execution of Petri nets.

Numerous researchers have reported progress towards successful integrations of neural networks and Petri net models. The advantage of incorporating artificial neural networks into Petri nets is mainly focused on developing the learning capability in Petri nets models. In Alsons fuzzy neural Petri net model, the weights and thresholds were not fuzzy. Hanna et al.’s fuzzy Petri nets combined neural networks with fuzzy Petri nets rather than modeled neural networks through the use of Petri net. Moreover, no fuzzy set was involved in their paper. Pedrycz et al. proposed a generalized fuzzy Petri net model but only crisp weights and thresholds are taken into consideration. Dunyak et al. proposed a practical algorithm for training fuzzy neural networks with fuzzy weights, inputs and outputs.

In this paper, we propose a unified formalism to manage both symbolic and numerical information based on high-level fuzzy Petri nets (HLFPN). Fuzzy functions, fuzzy reasoning and fuzzy neural networks are integrated in HLFPN. In HLFPN, a fuzzy place carries information to describe the fuzzy variable and the fuzzy set of a fuzzy condition. An arc is labeled with a fuzzy weight to represent the strength of connection between places and transitions. A fuzzy set and a fuzzy truth-value are attached to an uncertain fuzzy token to model imprecision and uncertainty. We have identified six types of uncertain transition:
calculation transitions to compute functions with uncertain fuzzy inputs; inference transitions to perform fuzzy reasoning; neuron transitions to execute computations in neural networks; duplication transitions to duplicate a uncertain fuzzy token to several tokens carrying the same fuzzy sets and fuzzy truth values; aggregation transitions to combine several uncertain fuzzy tokens with the same fuzzy variable; and aggregation-duplication transitions to amalgamate aggregation transitions and duplication transitions.

The organization of this paper is as follows. Fuzzy systems that contain fuzzy functions, fuzzy reasoning and fuzzy neural networks are discussed in the next section. In section 3, a high-level Petri net formalism is proposed, and the mapping between fuzzy functions, fuzzy reasoning and fuzzy neural networks to HLFPN is described. In section 4, an algorithm based on hierarchical HLFPN is developed. An example is used to illustrate the proposed approach in section 5. Finally, a summary of our approach and its potential benefits are given in the section 6.

2. Fuzzy Systems

The distinction between imprecise and uncertain information can be best explained by the canonical form representation (i.e. a quadruple of attribute, object, value, confidence) proposed by Dubois and Prade. Imprecision implies the absence of a sharp boundary of the value component of the quadruple; whereas, uncertainty is related to the confidence component of the quadruple which is an indication of our reliance about the information.

In a complicated fuzzy system, fuzzy function, fuzzy reasoning and fuzzy neural networks may be involved at the same time. In this paper, the three formalisms are integrated in HLFPN. In this section, fuzzy functions, fuzzy reasoning and fuzzy neural networks are discussed how to handle uncertainty and fuzziness.

2.1. Fuzzy Functions

In engineering, mathematics, and the sciences, functions are ubiquitous elements in modeling. Although a system can be modeled by a mathematical function, the inputs with uncertainty and fuzziness usually result in uncertain fuzzy outputs. To represent uncertain fuzzy numbers, we use a fuzzy number with a fuzzy truth-value as a truth-qualified fuzzy number, denoted as

\[(F, \tau)\]  

where \(F\) is a fuzzy set in a universe of discourse \(U\) and \(\tau\) is a fuzzy truth-value. The fuzzy set is to represent the intended meaning of fuzzy number; while, the fuzzy truth-value serves as the representation of uncertainty for its capability to express the possibility of the degree of truth.

To deal with the functions with uncertain fuzzy inputs and outputs, the extension principle developed by Zadeh and later elaborated by Yager enables us to extend the domain of a function on fuzzy sets. Let a function \(Y = f(X_1, X_2, \ldots, X_n)\) be a mathematical model of a subsystem, where \(X_1, X_2, \ldots, X_n\)
are fuzzy independent variables and their values are defined on the universes of discourse \(U_1, U_2, \ldots, U_n\), and \(Y\) is fuzzy dependent variable and its values are defined on the universe of discourse \(V\). If the inputs are \((F_1, \tau_1), (F_2, \tau_2), \ldots, (F_n, \tau_n)\), where fuzzy sets \(F_1, F_2, \ldots, F_n\), and \(\tau_i\) are defined on \(U_1, U_2, \ldots, U_n\), and \(T\), respectively, the membership function of the fuzzy output \(G\) is defined by the extension principle, that is

\[
\mu_G(v) = \sup_{u_1, u_2, \ldots, u_n} \{\min[\mu_{F_1}(u_1), \mu_{F_2}(u_2), \ldots, \mu_{F_n}(u_n)]\}
\]

(2)

where \(F_i^+ (i = 1 \sim n)\) is computed by a translation rule \(21\) \(\mu_{F_i}^+(u_i) = \mu_{\tau_i}(\mu_{F_i}(u_i))\).

Dong et al. \(4\) developed a procedure known as the vertex method to greatly simplify manipulations of the extension principle for continuous-valued fuzzy variables. The method was based on a combination of the \(a\)-cut concept and standard interval analysis. That is, a \(\alpha\)-cut of the output \(G\) is calculated by \(G^\alpha = f(F_1^\alpha, F_2^\alpha, \ldots, F_n^\alpha)\).

2.2. Fuzzy Reasoning

Expert systems are computer programs that emulate the reasoning process of a human expert or perform in an expert manner. They typically reason with uncertain and imprecise information. To represent uncertain imprecise information, a fuzzy proposition with a fuzzy valuation has been chosen by the authors \(11\), denoted as

\[(r, \tau)\]

(3)

where \(r\) is a fuzzy proposition of the form \("X is F"\) (i.e. \(X\) is a linguistic variable \(20\) and \(F\) is a fuzzy set in a universe \(U\)), and \(\tau\) is a fuzzy valuation. It should be noted that, for every formula \((r, \tau)\) (called a truth-qualified fuzzy proposition), we assume \(\tau \geq \tau(r|\pi)\) (i.e. \(\tau(r|\pi)\) is the real fuzzy truth value derived from \(r\) and a possibility distribution \(\pi\)), which means \(\mu_{\tau}(t)\) is the upper bound of the possibility that \(r\) is true to a degree \(t\). The fuzzy set is to represent the intended meaning of imprecise information; while, the fuzzy truth value serves as the representation of uncertainty for its capability to express the possibility of the degree of truth.

A general formulation of the inference rule for truth-qualified fuzzy propositions is expressed as follows.

\[
\begin{array}{c}
(r_1 \land r_2 \land \cdots \land r_n) \rightarrow q, \tau_1 \\
r'_1, \tau_2 \\
r'_2, \tau_3 \\
\vdots \\
r'_n, \tau_{n+1} \\
q', \tau_{n+2}
\end{array}
\]

(4)

where \(r_i, r'_i (i = 1 \sim n)\), \(q\), and \(q'\) are fuzzy propositions and characterized by \("X_i is \(F_i^\prime\)\", \("X_i is \(F_i^\prime\)\", \("Y is \(G^\prime\)\", and \("Y is \(G^\prime\)\", respectively; \(\tau_j (j = 1 \sim n+2)\) are
fuzzy valuations for truth values and defined by $\mu_{F_i}(t)$. $F_i$ and $F'_i$ are the subsets of $U_i$, while $G$ and $G'$ are the subsets of $V$. To derive $q'$ and $\tau_{n+2}$ of equation (4), we have proposed an approach based on possibility entailment, which three major steps were involved as follows 12.

- The fuzzy rules and fuzzy facts with fuzzy truth-values are transformed into a set of uncertain classical propositions with necessity and possibility measures.

- The possibilistic entailment is performed on the set of uncertain classical propositions.

- We reverse the process in the first step to synthesize all the classical sets obtained in the second step into a fuzzy set, and to compose necessity and possibility pairs to form a fuzzy truth-value.

Several inferred conclusions with the same linguistic variables should be aggregated. There are three steps involved in the aggregation 11: (1) the fuzzy rules and fuzzy facts with fuzzy truth-values are transformed into a set of uncertain classical propositions with necessity and possibility measures. (2) The aggregation of classical propositions with necessity and possibility measures is performed on the set of uncertain classical propositions. (3) We reverse the process in the first step to synthesize all the classical sets obtained in the second step into a fuzzy set, and to compose necessity and possibility pairs to form a fuzzy truth-value.

### 2.3. Fuzzy Neural Networks

Neural Networks, inspired by biological nervous systems, are a kind of connectionist architectures to replace symbolically structured representations with distributed representations in the form of weights between a massive set of interconnected neurons 8. Fuzzy neural networks indicate the conventional neural networks with fuzzy inputs and weights. In this paper, the uncertain fuzzy inputs are considered and denoted as a truth-qualified fuzzy number $(F_i, \tau_i)$, where $F_i$ is a fuzzy set and $\tau_i$ is a fuzzy truth value, and the fuzzy weights is indicated as fuzzy sets $W_i$ in the universes of discourse $W$. As shown in Figure 1, the threshold value $\theta$ is substituted by $-W_0$ and a feedforward fuzzy neuron model is characterized as follows.

$$ G = f(\sum_{i=1}^{n} F_{i}^{+} W_i + W_0) $$

(5)

where $f(\cdot)$ is a transfer function and $F_{i}^{+}(i = 1 \sim n)$ is computed by a translation rule $21 \mu_{F_{i}^{+}}(u_i) = \mu_{\tau_i}(\mu_{F_{i}^{+}}(u_i))$. The extension principle is then applied to $f(\cdot)$ and the membership function of the output $G$ is calculated as follows.

$$ \mu_G(v) = \text{Sup}_{u=f(\sum_{i=1}^{n} u_i w_i + W_0)} \{ \min[\mu_{W_0}(w_0), \mu_{W_1}(w_1), \ldots, \mu_{W_n}(w_n)], \mu_{F_{1}^{+}}(u_1), \mu_{F_{2}^{+}}(u_2), \ldots, \mu_{F_{n}^{+}}(u_n)] \} $$

(6)
Dunyak J. et al.\textsuperscript{6} pointed out that if $f(\cdot)$ is a strictly increasing function, the computational complexity of equation (6) can be largely reduced by applying vertex method and interval analysis. That is, the $\alpha$-cut of output $G$ is computed by the following equation.

$$G^\alpha = [G_1^\alpha, G_2^\alpha]$$

$$= \left[ f(\sum_{i=1}^{n} \min(F_i^\alpha W_i^\alpha, F_i^\alpha W_i^\alpha, F_i^\alpha W_i^\alpha) + W_0^\alpha), \right.$$  
$$\left. f(\sum_{i=1}^{n} \max(F_i^\alpha W_i^\alpha, F_i^\alpha W_i^\alpha, F_i^\alpha W_i^\alpha) + W_0^\alpha) \right]$$

(7)

where $G_1^\alpha$ and $G_2^\alpha$ indicate the left and right endpoints of $\alpha$-cut of $G$, respectively.

The fuzzy weights are updated by training data pairs $(F_i, G'_i)$ and the backpropagation learning rule, that is

$$W_i(new) = W(old) - \eta \cdot \nabla W_i e$$

(8)

where $\eta$ is a learning rate and $\nabla W_i e$ is the gradient of the error function $e$. In this paper, a quadratic error criterion\textsuperscript{6} is adopted as the error function, denoted as

$$e = \frac{1}{2n} \sum_{i=1}^{n} [w_{i1}(G_1^\alpha - G_1^\alpha)^2 + w_{i2}(G_2^\alpha - G_2^\alpha)^2]$$

(9)

where $G_1^\alpha$ and $G_2^\alpha$ indicate the left and right endpoints of $\alpha$-cut of $G$, respectively, and $w_{ij}$ is a relative weight of the importance of different $\alpha$ levels.

### 3. High-level Fuzzy Petri Nets

To integrate several modeling formalisms into a complicated system, a more general representation model is required, which is capable of managing symbolic and numerical information simultaneously and able to synthesize diverse modeling tools. Petri nets offer the possibility to achieve this goal because of its flexibly graphic representation, and its capability of formal analysis.
3.1. Petri Nets

A Petri net is a directed, weighted, bipartite graph consisting of two kinds of nodes, called places \((P_i)\) and transitions \((t_j)\), where arcs are either from a place to a transition or from a transition to a place\(^{16}\). Murata has formally defined Petri nets as a 5-tuple\(^{14}\): \(PN = (P, T, F, W, M_0)\), where \(P = \{P_1, P_2, \ldots, P_m\}\) is a finite set of places, \(T = \{t_1, t_2, \ldots, t_n\}\) is a finite set of transitions, \(F \subseteq (P \times T) \cup (T \times P)\) is a set of arcs, \(W : F \rightarrow \{1, 2, 3, \ldots\}\) is a weight function, and \(M_0 : P \rightarrow \{0, 1, 2, 3, \ldots\}\) is the initial marking. A marking \(M\) is a \(m\)-vector, \(\{M(P_1), \ldots, M(P_m)\}\), where \(M(P_i)\) denotes the number of the tokens in place \(P_i\). The incidence matrix \(A = [a_{ij}]\) is a \(n \times m\) matrix of integers and its typical entry is defined by \(a_{ij} = a_i^+ - a_i^−\), where \(a_i^+\) is the weight of the arc from a transition \(t_i\) to its output place \(P_j\), and \(a_i^−\) is the weight of the arc to a transition \(t_i\) from its input place \(P_j\). The reachability set \(R(M_0)\) of a Petri net is defined as the set of all possible markings reachable from \(M_0\). A place having two or more than two output transitions is referred to as a conflict. Two transitions are said to be concurrent if they are causally independent. The evolution of markings, used to simulate the dynamic behavior of a system, is based on the firing rule, such as: a transition \(t\) is enabled if each input place \(i\) is marked with at least \(w(p, t)\) tokens, where \(w(p, t)\) is the weight of the arc from \(p\) to \(t\); an enabled transition may or may not be enabled; and a firing of an enabled transition \(t\) removes \(w(p, t)\) tokens from each input place \(p\) of \(t\), and adds \(w(t, p)\) tokens to each output place \(p\) of \(t\), where \(w(t, p)\) is the weight of the arc from \(t\) to \(p\). Some notations are introduced as follows: \(\bullet t_j\) denotes the input places of \(t_j\), \(t_j \bullet\) denotes the output places of \(t_j\), \(\bullet P_i\) denotes the input transitions of \(P_i\), and \(P_i \bullet\) denotes the output transitions of \(P_i\).

3.2. High-level Fuzzy Petri Nets

In this paper, we propose a high-level fuzzy Petri nets (HLFPN) to integrate fuzzy functions, fuzzy reasoning, and fuzzy neural networks. The HLFPN is defined as follows.

**Definition 1 (High-level Fuzzy Petri nets):** A high-level fuzzy Petri net HLFPN is defined as a five-tuple,

\[
HLFPN = (FP, UT, A, W, \Theta, M_0)
\]

\(FP = \{P_1(X_1, F_1), P_2(X_2, F_2), \ldots, P_m(X_m, F_m)\}\) is a finite set of fuzzy places, where \(P_i\) represents a fuzzy place, \(X_i\) indicates a fuzzy variable and \(F_i\) is a fuzzy subset of \(U_i\) to represent the fuzzy set of \(X_i\). \(UT = \{(t_1, \tau_1), (t_2, \tau_2), \ldots, (t_n, \tau_n)\}\) is a finite set of uncertain transitions, where \(t_j\) represents the connectivity of fuzzy places and \(\tau_j\) is a fuzzy truth-value to represent the uncertainty about the connectivity between fuzzy places. \(A\) is a set of arcs. \(W = \{W_1, W_2, \ldots, W_k\}\) is a finite set of fuzzy weights which are attached to the relevant input arcs of uncertain transitions to represent the connection strengths between fuzzy places and uncertain transitions.
\[ M_0 = \{ M(P_1), M(P_2), \ldots, M(P_m) \} \]

The fuzzy truth value serves as the representation of uncertainty for its capability to express the possibility of the degree of truth. If a fuzzy proposition is attached with a fuzzy truth value, we call it a truth-qualified fuzzy proposition. In definition 1, we assume that \( \tau_i(t) \) (for \( t \in [0, 1] \)) of each \( \tau_i(i = 1 \sim m + n) \) means the upper bound of the possibility measure that the degree of truth is \( t \).

Each token is associated with a pair of fuzzy sets \( X_i(F_i', \tau_i) \) (called an uncertain fuzzy token). Fuzzy tokens are inserted into the related fuzzy places that have the same fuzzy variables. To simulate the dynamic behavior of a fuzzy system, a marking in a high-level fuzzy Petri net is changed according to the firing rule: a firing of an enabled uncertain transition \( t_j \) removes the uncertain fuzzy token from each fuzzy input place \( P_i \) of \( t_j \), adds a new token to each output place \( P_k \) of \( t_j \), and the fuzzy sets and fuzzy truth value attached to the new token will be computed based on the mechanisms of the type of \( t_j \). A simple example of \( HLFPN \) is illustrated in Figure 2.

We have identified six types of uncertain transition: calculation transitions to compute fuzzy functions; inference transitions to perform fuzzy reasoning; neuron transitions to execute computations in neural networks; duplication transitions to duplicate a uncertain fuzzy token to several tokens carrying the same fuzzy sets and fuzzy truth values; aggregation transitions to combine several uncertain fuzzy tokens with the same variable; and aggregation-duplication transitions to combine aggregation transitions and duplication transitions.

**Type 1: Calculation transition \( t^c \).** A calculation transition represents a mathematical function. Its fuzzy input places indicate the fuzzy independent variables of the function; meanwhile, its fuzzy output places represent the fuzzy depen-
High-Level Fuzzy Petri Nets for Managing Symbolic and Numerical Information

Figure 3: Modeling fuzzy calculation through HLFPN: (a) before firing calculation transition \( t_1 \), (b) after firing \( t_1 \).

Figure 4: Modeling fuzzy rule-based reasoning through HLFPN: (a) before firing inference transition \( t_1 \), (b) after firing \( t_1 \).

defined variables of the function. In Figure 3, after firing the calculation transition \( t_1 \), the tokens will be removed from the input places of \( t_1 \), a new token will be inserted into the output place of \( t_1 \), and the output fuzzy set attached to the new token are computed by the extension principle or vertex method (See section 2.1). It should be noted that in Figure 3, \( F_i \), \( G \), \( W_i \), \( \tau_1 \), \( \theta \) are nullities, that is, they do not participate in the calculation.

Type 2: Inference transition \((t^i)\). An inference transition serves as modeling of a truth-qualified fuzzy rule. A truth-qualified fuzzy rule having multiple antecedents is represented as

\[
(r_1 \land r_2 \land \cdots \land r_n) \rightarrow q, \tau_1
\]

where \( r_i \) and \( q \) are of the forms of “\( X_i \) is \( F_i \)” and “\( Y \) is \( G \)”, respectively. In Figure 4, after firing the inference transition \( t_1 \), the tokens will be removed from the input places of \( t_1 \), a new token will be deposited into the output place of \( t_1 \), and the fuzzy set and the fuzzy truth value attached to the new token are derived by three steps: (i) Transformation; (ii) Inference; (iii) Composition (see section 2.2.). It should be
noted that in Figure 4, $W_i, \theta$ are nullities, that is, they do not participate in the fuzzy reasoning.

**Type 3: Neuron transition** ($t^n$). A neuron transition is used to model fuzzy neural networks and connects its inputs and outputs. Figure 5 is a generalized Petri net model of Figure 1. In Figure 5, after firing the neuron transition $t^n_1$, the tokens will be removed from the input places of $t^n_1$, a new token will be deposited into the output place of $t^n_1$, and the output fuzzy set attached to the new token are derived by transfer function. (see section 2.3). It should be noted that in Figure 5 $F_i, G$ and $\tau_1$ are nullities, that is, they do not participate in the computation of fuzzy neural networks.

**Type 4: Aggregation transition** ($t^a$). An aggregation transition is used to aggregate the conclusions of several truth-qualified fuzzy rules which have the same linguistic variable, and to link the antecedent of a truth-qualified fuzzy rule which also have the same variable. For example, there are $m$ truth-qualified fuzzy rules
Figure 7: Modeling the duplication of an uncertain fuzzy token through HLFPN: (a) before firing the duplication transition \( t^d_1 \), (b) after firing \( t^d_1 \).

having the same variable in the conclusions, denoted as

\[
(r_1 \to q_{11}, \tau_1), (r_2 \to q_{12}, \tau_2), \ldots, (r_m \to q_{1m}, \tau_m)
\]

where \( q_{1i} \) is “\( Y \) is \( G_{1i} \)”.

Type 5: Duplication transition \( (t^d) \). The purpose of duplication transitions is to avoid the conflict by duplicating the token. For example, there are \( m \) truth-qualified fuzzy rules having the same linguistic variable in the antecedents, denoted as

\[
(r_{11} \to q_1, \tau_1), (r_{12} \to q_2, \tau_2), \ldots, (r_{1l} \to q_1, \tau_l)
\]

where \( r_{1i} \) means “\( X_i \) is \( F_{1i} \)”.

Type 6: Aggregation-duplication transition \( (t^{ad}) \). An aggregation-duplication transition is a combination of an aggregation transition and a duplication transition (see Figure 7). It is used to link all fuzzy propositions which have the fuzzy rules having the same variable in the conclusions and \( l \) truth-qualified fuzzy rules having the same variable in the antecedents, denoted as

\[
(r_1 \to q_{11}, \tau_1), (r_2 \to q_{12}, \tau_2), \ldots, (r_m \to q_{1m}, \tau_m), \\
(q_{1(m+1)} \to s_1, \tau_{m+1}), (q_{1(m+2)} \to s_2, \tau_{m+2}), \ldots, (q_{1(l+m)} \to s_l, \tau_{m+l})
\]
Figure 8: An aggregation-duplication transition is a combination of aggregation transition and a duplication transition.

where \( q_{1i} \) is of the form of “\( Y_1 \) is \( G_{1i} \)”. They are linked by an aggregation-duplication transition shown in Figure 8. After firing the aggregation-duplication transition \( t_{1}^{ad} \), the tokens in the input places of \( t_{1}^{ad} \) will be removed, new tokens will be deposited into the output places of \( t_{1}^{ad} \), and the fuzzy sets and the fuzzy truth values attached to the new tokens are derived by three steps: (i) Transformation; (ii) Aggregation; and (iii) Composition. It should be noted that in Figure 3, \( Wi, \tau_i, \theta \) are nullities, that is, they do not participate in the calculation.

3.3. Analysis of Fuzzy Petri Nets

This section describes how fuzzy Petri nets can be analyzed. Two major Petri net analysis methods, the coverability tree and state equation, are used to analyze fuzzy Petri nets.

The Coverability Tree The coverability tree represents the reachability set of a fuzzy Petri net. Given a fuzzy Petri net, a tree representation of the markings can be constructed. In this tree, a symbol \( \omega \) is used to represent “infinity”, nodes represent markings reachable from \( M_0 \) and each arc represents an uncertain transition firing that transforms one marking to another. Some of the behavioral properties that can be studied by using the coverability tree are boundedness, safeness and deadlock in uncertain transitions. For a bounded fuzzy Petri net, the coverability tree is called the reachability tree.

State Equation The state equation that governs the dynamic behavior of concurrent fuzzy systems modeled by fuzzy Petri nets is represented by \( A^T x = \Delta M \), where \( \Delta M = M_n - M_0 \), \( A \) is the incidence matrix and \( x \) is an \( n \times 1 \) column vector called the firing count vector. The \( i \)th entry of \( x \) denotes the number of times that uncertain transition \( t_i \) must fire to transform \( M_0 \) to \( M_n \). The state equation is used
to solve the reachability problem, that is, the problem of finding if $M_n \in R(M_0)$ for a given $M_n$. If $M_n$ is reachable from $M_0$, then state equation has a solution in nonnegative integers. If the state equation has no solution, then $M_n$ is not reachable from $M_0$.

3.4. Fuzzy Functions and Fuzzy Petri Nets

The algorithmic flow chart of functions can be clearly displayed through the use of graphic representation such as the proposed high-level fuzzy Petri nets. The mapping between fuzzy functions and $HLFPN$ is described as follows.

- **Fuzzy places**: Fuzzy places correspond to fuzzy variables of a fuzzy function. Fuzzy input and fuzzy output places of a calculation transition are used to represent the fuzzy independent and dependent variables of a fuzzy function, respectively.

- **Uncertain fuzzy tokens**: An uncertain fuzzy token represents a truth-qualified fuzzy number. The fuzzy sets and fuzzy truth-values are attached to tokens to represent the values of fuzzy independent variables and our confidence level about the fuzzy numbers, respectively.

- **Calculation transitions**: A calculation transition represents a mathematical function. Once a calculation transition is fired, the computation of the fuzzy output number is carried out through the use the extension principle. An aggregation transitions are designed to aggregate the different values of a fuzzy dependent variable, a duplication transition is used to duplicate uncertain fuzzy tokens to avoid the conflict problem, and an aggregation-duplication transition links the fuzzy places with the same fuzzy variables.

3.5. Fuzzy Reasoning and Fuzzy Petri Nets

Many researchers have devoted themselves into modeling fuzzy rule-based reasoning via fuzzy Petri nets $^{2,3,9,13,17,18}$. There are several rationales behind to base a computational paradigm for fuzzy reasoning on Petri net theory:

- Petri nets achieve the structuring of knowledge within rule bases, which can express the relationships among rules and help experts construct and modify rule bases.

- Petri net’s graphic nature provides the visualization of dynamic behavior of rule-based reasoning.

- Petri nets make it easier to design efficient reasoning algorithm.

- Petri net’s analytic capability provides a basis for developing knowledge verification technique.
Petri nets can model the underlying relationship of concurrency among rules activation, which is an important aspect where real-time performance is crucial.

The three key components in fuzzy rule-based reasoning: fuzzy propositions, fuzzy rules and fuzzy facts, can be formulated as places, transitions and tokens, respectively. The mapping between fuzzy reasoning and HLFPN is described as follows.

- **Fuzzy places:** Fuzzy places correspond to fuzzy propositions. The linguistic variables and fuzzy sets, attached to the fuzzy places, represent the variables of fuzzy propositions and the corresponding values. Fuzzy input and fuzzy output places of a truth-qualified transition are used to represent the antecedent and conclusion parts of a truth-qualified fuzzy rule, respectively.

- **Uncertain fuzzy tokens:** An uncertain fuzzy token represents a truth-qualified fuzzy fact. The fuzzy sets and fuzzy truth-values are attached to tokens to represent the values and our confidence level about the observed facts, respectively.

- **Inference transitions:** An inference transition represents a truth-qualified fuzzy rule. An aggregation transitions are used to aggregate the conclusion parts of rules that have the same linguistic variables, a duplication transition duplicates uncertain fuzzy tokens, and an aggregation-duplication transition links the fuzzy propositions with the same linguistic variables.

### 3.6. Fuzzy Neural Networks and Fuzzy Petri Nets

The advantage of incorporating artificial neural networks into Petri nets is mainly focused on developing the learning capability in Petri nets models. Numerous researchers have reported progress towards successful integrations of neural networks and Petri net models\(^1\,^7\,^{15}\). The mapping between fuzzy neural networks and HLFPN is described as follows.

- **Fuzzy places:** Fuzzy places correspond to fuzzy variables of fuzzy neural networks. Fuzzy input and fuzzy output places of a neuron transition are used to represent the fuzzy input and output variables of a neural network, respectively.

- **Uncertain fuzzy tokens:** An uncertain fuzzy token represents a truth-qualified fuzzy number. The fuzzy sets and fuzzy truth-values are attached to tokens to represent the values of fuzzy input/output and our confidence level about the fuzzy numbers, respectively.

- **Neuron transitions:** A neuron transition represents a fuzzy transfer function. Once a neuron transition is fired, the computation of the fuzzy transfer
function is performed by the extension principle. An aggregation transition is designed to aggregate the different values of a fuzzy variable, a duplication transition is used to duplicate uncertain fuzzy tokens, and an aggregation-duplication transition links the fuzzy places with the same fuzzy variables.

3.7. **Hierarchical Structures of HLFPN**

To overcome the complexity arising from large sizes of HLFPN, a hierarchical structure of HLFPN is developed. In a hierarchical structure, HLFPN is divided into several hierarchies that contain smaller HLFPN and may or may not include another hierarchy. As illustrated in Figure 9(a), a hierarchy is drawn as a double-lined square to connect fuzzy places and viewed as a transition. The hierarchical HLFPN in Figure 9(a) is considered as a main hierarchy $H_0$ and contains three hierarchies, $H_1$, $H_2$ and $H_3$. In this example, the main hierarchy $H_0$ is at the top level of the hierarchical structure, and $H_0$, $H_2$ and $H_3$ are at the second level. The connections between hierarchies are achieved by defining importing and exporting fuzzy places (see Figure 9). That is, an exporting fuzzy place w.r.t. a hierarchy is defined as a fuzzy place that is connected to the hierarchy by an arc from the fuzzy place to the hierarchy; meanwhile, an importing fuzzy place w.r.t. a hierarchy is defined as a fuzzy place connected to the hierarchy by an arc from the hierarchy to the fuzzy place. In this hierarchical HLFPN, fuzzy places $P_1$ and $P_2$ are the exporting fuzzy place w.r.t. hierarchy $H_1$, and fuzzy place $P_4$ is the importing fuzzy places w.r.t. hierarchy $H_1$. In the hierarchy $H_1$, fuzzy places $P_1$ and $P_2$ are the importing fuzzy places w.r.t. $H_0$, and fuzzy place $P_5$ is the exporting fuzzy place w.r.t. $H_0$.

When a token is inserted into the fuzzy place $P_1$ in $H_0$, it will be transited into hierarchy $H_1$ and added to place $P_1$ in hierarchy $H_1$. Similarly, once a token enters into place $P_2$ in $H_0$, it will be sent into hierarchy $H_1$ and reaches the fuzzy place $P_2$ in $H_1$. After firing the transitions in $H_1$, the computations of fuzzy reasoning are triggered and the token arrives the fuzzy place $P_5$ in $H_1$ and then enters the fuzzy place $P_4$.

There are two main benefits by having a hierarchical structure for HLFPN: (1) the notion of hierarchy makes easy the handling of complex systems through decomposition; and (2) a hierarchical Petri net facilitates the reusability, namely, each hierarchy can be considered as a reuse unit.

4. Algorithm

To consider the efficiency of HLFPN, an algorithm of transition firing is proposed based on the notion of extended fuzzy markings. An extended fuzzy marking, denoted as $FM^E$, is a $k$-vector, where $k$ is the number of uncertain fuzzy tokens. The elements of $FM^E$ denoted by $FM^E(P_i)$ are called extended fuzzy places, defined as

$$FM^E(P_i) = [P_i, X_i, F_i, \tau_i, P_i\bullet, \bullet(P_i\bullet)\setminus\{P_i\}, (P_i\bullet)\bullet]$$

(15)

where fuzzy truth value attached to the token in fuzzy place $P_i$; $P_i\bullet$ is the output
transition of $P_i$. From an extended fuzzy places $FM^E(P_i)$, we know that: (a) the fuzzy set and the fuzzy truth value attached to the token in $P_i$ (i.e. $F'_i$ and $\tau_i$), (b) the other tokens needed to fire $P_i$ (i.e. $\bullet(P_i)\backslash\{P_i\}$), (c) what kind of computation to carry out after firing (i.e. the type of $P_i$), and (d) where to go for the new tokens after firing (i.e. $(P_i)\bullet$). The algorithm is used to manage the evolution of extended fuzzy marking. It is described as follows:

Algorithm 1 (High-level Fuzzy Petri Nets)

1. Get the initial extended fuzzy marking $FM^E_0$, which consists of all source fuzzy places.

2. For each $i$, set a current extended fuzzy marking $FM^E_i = FM^E_i$, and the next extended fuzzy marking $FM^E_{i+1} = \emptyset$.

3. Select an element of the current extended fuzzy marking, $FM^E_i(P_j) = [P_j, X_j, F'_j, \tau_j, P_i \bullet, \bullet(P_i)\backslash\{P_j\}, (P_i)\bullet].$

4. (a) If $P_i \bullet$ is a calculation transition, and the extended fuzzy place of each $P_i \in \bullet(P_i)\backslash\{P_i\}$ exists in $FM^E_k$, then infer the extended fuzzy place $FM^E_{i+1}(P_k)$ of $P_k = (P_i)\bullet$ by extension principle.

(b) Else if $P_i \bullet$ is an inference transition, and the extended fuzzy place of each $P_i \in \bullet(P_i)\backslash\{P_i\}$ exists in $FM^E_k$, then infer the extended fuzzy place $FM^E_{i+1}(P_k)$ of $P_k = (P_i)\bullet$ by (i) transformation, (ii) inference, and (iii) composition.

(c) Else if $P_i \bullet$ is a neuron transition, and the extended fuzzy place of each $P_i \in \bullet(P_i)\backslash\{P_i\}$ exists in $FM^E_k$, then infer the extended fuzzy place $FM^E_{i+1}(P_k)$ of $P_k = (P_i)\bullet$ by computing transfer function.

(d) Else if $P_i \bullet$ is a duplication transition, then infer the extended fuzzy place $FM^E_{i+1}(P_k)$ of each $P_k \in (P_i)\bullet$ by duplication.

(e) Else if $P_i \bullet$ is an aggregation transition, and the extended fuzzy place of each $P_i \in \bullet(P_i)\backslash\{P_i\}$ exists in $FM^E_k$, then infer the extended fuzzy place $FM^E_{i+1}(P_k)$ of $P_k = (P_i)\bullet$ by (i) transformation, (ii) aggregation, and (iii) composition.

(f) Else if $P_i \bullet$ is an aggregation-duplication transition, and the extended fuzzy place of each $P_i \in \bullet(P_i)\backslash\{P_i\}$ exists in $FM^E_k$, then infer the extended fuzzy place $FM^E_{i+1}(P_k)$ of each $P_k \in (P_i)\bullet$ by (i) transformation, (ii) aggregation, and (iii) composition.

5. (a) If the output transition of $P_i$ is fired, then insert the inferred extended fuzzy place $FM^E_{i+1}(P_k)$ into the next extended fuzzy marking $FM^E_{i+1}$.

(b) Else if the output transition of $P_i$ is a hierarchy, then insert the extended fuzzy place $FM^E(P_i)$ into the hierarchy and wait for the final extended fuzzy marking of the hierarchy to be inserted into the current extended fuzzy marking.

(c) Else insert this element $FM^E(P_i)$ and each $FM^E(P_i)\backslash\{P_i\}$ into the next extended fuzzy marking $FM^E_{i+1}$.

6. Delete the element $FM^E(P_i)$ and each $FM^E(P_i)\backslash\{P_i\}$ from the current extended fuzzy marking.

7. Repeat step 3 to step 6 until no element is in the current extended fuzzy marking.

8. Repeat step 2 to step 7 until all output transitions in the current extended fuzzy marking are not fired.

9. Send the final extended fuzzy marking to the upper-leveled hierarchy.
Figure 9: (a) Hierarchical structure of HLFPN; (b) hierarchy $H_1$: modeling fuzzy reasoning; (c) hierarchy $H_2$: modeling fuzzy neural network; (d) hierarchy $H_3$: modeling fuzzy function.

5. Example

A HLFPN in Figure 9 is illustrated how to apply the proposed approach and algorithm to manage symbolic and numerical information and integrate fuzzy function, fuzzy reasoning and fuzzy neural network. This HLFPN contains three hierarchies $H_1, H_2$ and $H_3$. Hierarchy $H_1$ represents two truth-qualified fuzzy rules:

Rule 1: (*IF* $X_1$ is $\tilde{F}_{13}$ *THEN* $X_4$ is $\tilde{F}_{42}$, true)

Rule 2: (*IF* $X_2$ is $\tilde{F}_{23}$ *THEN* $X_4$ is very $\tilde{F}_{43}$, very true)

where $X_i$ is a linguistic variable and $F_i$ is a fuzzy subset of $U_i$.

Hierarchy $H_2$ models a two-layered fuzzy neural network (see Figure 10).

Hierarchy $H_3$ indicates a function, that is $X_6 = f(X_3) = X_3^2$. The detailed information about this HFLPN is summarized in Table 1.

The input data is listed as follows.
Symbolic data:

(1) It is very true that $X_1$ is more or less $F_{12}$. $< X_1(\sqrt{F_{12}}, \tau^2) >$
(2) It is true that $X_2$ is very $F_{22}$. $< X_2(F_{22}^2, \tau) >$

Numerical data:

It is very true that $X_3$ is more or less about 2. $< X_3(\sqrt{F_{34}}, \tau^2) >$

where fuzzy set "about 2" and fuzzy truth-value true is denoted as $F_{34}$ and $\tau$, respectively.

After performing the proposed algorithm, the outputs of the HLFPN are:

Output 1: (It is true that $X_5$ is $F_{52} = (\{0, 0.924, 0.5, 0.1, 0.5, 0.9\}, 0.89))$

Output 1: (It is true that $X_6$ is $F_{63} = (\{0, 0.5, 0.5, 1, 0.5, 0\}, 0.89))$

6. Conclusion

A unified formalism to manage both symbolic and numerical information has been proposed based on high-level fuzzy Petri nets (HLFPN). Major features of HLFPN are as follows. (1) A fuzzy place carries information to describe the fuzzy variable and the fuzzy set of a fuzzy condition; (2) an arc is labeled with a fuzzy weight to represent the strength of connection between places and transitions; (3) an uncertain fuzzy token represents symbolic or numerical data with imprecision and uncertainty; (4) six types of uncertain transition are introduced to fulfill the mechanisms of diverse subsystems; and (5) a hierarchical structure of HLFPN and an algorithm to execute HLFPN have also been presented. The proposed HFLPN offers several benefits:

- It offers a basis for managing symbolic and numerical information.
- It can integrate fuzzy functions, fuzzy reasoning and fuzzy neural networks into one unified formalism in which uncertainty and fuzziness can be taken into account.


<table>
<thead>
<tr>
<th>Hierarchy</th>
<th>Fuzzy Place</th>
<th>Uncertain Transition</th>
<th>Fuzzy Weight</th>
<th>Fuzzy Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_0$</td>
<td>$P_1(X_1, X_{13}) = N$; $t_1^0, t_{13} = N$; $W_1 = N$; $\theta_1 = N$;</td>
<td>$P_2(X_2, X_{21}) = N$; $H_1$; $H_2$; $H_3$</td>
<td>$W_2 = N$; $\theta_2 = N$;</td>
<td>$W_3 = N$; $\theta_3 = N$;</td>
</tr>
<tr>
<td></td>
<td>$P_3(X_3, X_{33}) = N$; $P_4(X_4, X_{41}) = N$; $P_5(X_5, X_{52}) = N$; $P_6(X_6, X_{63}) = N$</td>
<td></td>
<td>$W_4 = N$; $\theta_4 = N$;</td>
<td></td>
</tr>
</tbody>
</table>

| $H_1$     | $P_1(X_1, X_{13}) = N$; $t_1^i, \tau_i$; $W_1 = N$; $\theta_1 = N$; | $P_2(X_2, X_{23}) = N$; $t_2^i, \tau_i^2$; $W_2 = N$; $\theta_2 = N$; | $W_3 = N$; $\theta_3 = N$; | $W_4 = N$ |

| $H_2$     | $P_1(X_1, X_{15}) = N$; $t_1^i, \tau_{12} = N$; $W_1 = N$; $\theta_1 = N$; | $P_2(X_2, X_{25}) = N$; $t_2^i, \tau_2 = N$; $W_2 = N$; $\theta_2 = N$; | $W_3 = "about 0.7"$; $\theta_3 = "about 0.5"$; | $W_4 = "about 0.7"$; $\theta_4 = "about 0.4"$ |

| $H_3$     | $P_1(X_1, X_{15}) = N$; $t_1^i, \tau_{13} = N$; $W_1 = N$; $\theta_1 = N$; | $P_2(X_2, X_{25}) = N$; $t_2^i, \tau_3 = N$; $W_2 = N$; $\theta_2 = N$; | $W_3 = "about 0.6"$; $\theta_3 = "about 0.9"$; | $W_4 = "about 0.9"$; $\theta_4 = "about 0.8"$ |

Note: (1) $N$: nullity; (2) $F = F^2$; (3) more or less $F = \sqrt{F}$; (4) $\tau$: true; (5) "about $T" = \{\frac{1}{2}, \frac{3}{4}, \frac{5}{8}, \frac{7}{10}, \frac{9}{12}\}$; (6) "about $T" = \{\frac{1}{12}, \frac{3}{24}, \frac{5}{18}, \frac{7}{14}, \frac{9}{10}\}$.

- Its hierarchical structure makes easy the handling of complex systems through decomposition and facilitates the reusability.
- The efficiency of integrated systems can be improved by the proposed algorithm based on HLFPN.

Our future work is to integrate the mathematical modeling, the fuzzy reasoning, and fuzzy neural networks studies in a bridge damage assessment model based on HLFPN.

References