The Strategies of Optimizing Fuzzy Petri Nets by Using an Improved Genetic Algorithm

Li Yang¹, Yue Xiao-bo²

¹Faculty of Art and Design, Hunan International Business Vocational College, Changsha 410200, China
²Institute of Compu. & Commu. Engineering, Changsha Univ. of Sci. & Tech., Changsha 410015, China
*Corresponding author, e-mail: liyangpeace@sina.com

Abstract

It is very important for constructing a FPN (fuzzy petri net) to accurately find out all parameters of fuzzy production rules. In this paper, an improved genetic algorithm is introduced into the process of exploring the optimal parameters of a modified FPN. Realization of the algorithm does not depend on experiential data and requirements for the initial input of the FPN are not stringent. Simulation experiment shows that the parameters trained by the above algorithm are highly accurate and the FPN model constructed by these parameters possesses strong generalizing capability and self-adjusting purpose.

Keywords: FPN; fuzzy production rule; fuzzy reasoning; improved genetic algorithm

Copyright © 2016 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

Fuzzy petri nets both combine the capability of describing asynchronous concurrency and providing graphical representation (which makes knowledge representation simple and clear) and possess the capability of fuzzy reasoning of fuzzy systems (which favours the inference, analysis and verification of knowledge and decision-supporting), so the FPN is a kind of good modeling tool for knowledge base systems based on fuzzy production rules. However, weak self-learning capability is a defect of fuzzy systems. Fuzzy production rules express knowledge by means of IF-THEN structure, determining some parameters such as weight, threshold and CF (certainty factor) is dependent to a large extent upon someone’s experience, so these parameters are difficult to obtain accurately (sometimes even could not be obtained), which hinders knowledge reasoning and generalizing capability of FPNs. Nowadays in the computer science field, the study on searching and optimizing parameters of FPNs has tight restrictions and narrow scope of application in general [1-5]. For example, the FPN model based on a BP (Back Propagation) network (which adopts the thought of layering and could realize the optimization of all parameters including weight, threshold and CF) possesses strong generalizing capability and have improved its self-learning performance after being trained. However, in order to make the obtained parameters (after learning) closer to the ideal parameters, it needs analyzing the obtained parameters to make continuous adjustments to the initial input and requiring thresholds and CFs of each transition in the OR rule are all equal.

On the basis of this study, extensive research are carried out and presented in this paper. Firstly, traditional Genetic Algorithm (GA) is good at searching the globally optimal solution, that characteristic is fully utilized and finely improved in this paper, an algorithm of exploring the optimal parameters is proposed which is built on the basis of an continuous fuzzy reasoning function; secondly, scope application of the OR rule is expanded, which enable each transition to possess its own threshold and CF. Simulation experiment shows that the accuracy of the obtained parameters in terms of the strategies in this paper is better than that of other methods, and realization is easier and more convenient, so it is more suitable for practical needs.
2. FPN and Fuzzy Reasoning Function

2.1. FPN

A FPN model includes places, transitions, CFs, thresholds and weights, all of which are defined as a combination of eight elements [1].

**Definition 1 FPN**

\[ \text{FPN} = \{P, T, I, O, M, \tau, W, U\} \]

Among them, \( P = \{p_1, p_2, ..., p_n\} \) is a finite set of places;
\( T = \{t_1, t_2, ..., t_m\} \) is a finite set of transitions;
\( I(O): T \rightarrow P \) is an input (output) function, which reflects the mapping relation from one transition to one place;
\( M: P \rightarrow [0,1] \) is a mapping, every place node \( p_i \in P (i=1,2,...,n) \) has one flag value \( M(p_i) \), which reflects the truth degree of a proposition represented by the corresponding place node;
\( \tau = (\tau_1, \tau_2, ..., \tau_m) \) is the threshold of transition \( t_j (j=1,2,...,m) \);
\( W = \{w_1, w_2, ..., w_q\} \) is a weight collection of rules, which reflects the degree that the prerequisite of rules supports for the conclusion;
\( U = (u_1, u_2, ..., u_m) \), \( u_i \) is the CF of transition \( t_j , u_i \in [0,1] (j=1,2,...,m) \).

Fuzzy production rules are used to describe the fuzzy relation of all propositions, which are appropriate to express those fuzzy or uncertain knowledge. If \( R = \{R_1, R_2, ..., R_r\} \) is a system of fuzzy production rules, the formal definition of \( R_k (k=1,2,...,r) \) belongs to the following two categories in general.

**Definition 2**

1. Rule AND: \( R_k: \text{IF } d_1 \text{ and } d_2 \text{ and } ... \text{ and } d_n \text{ THEN } d, u, \tau, w_1, w_2, ..., w_n \).
   
   In the above rule, \( d_i \) is the premise proposition, \( d \) is the result proposition, \( u \) is the CF, \( \tau \) is the threshold, \( w_i \) is the weight (which meets \( 0 < w_i < 1 \) and \( \sum w_i = 1, i=1,2,...,n \) ), the corresponding FPN model shows in the figure 1;

2. Rule OR: \( R_k: \text{IF } d_1 \text{ or } d_2 \text{ or } ... \text{ or } d_n \text{ THEN } d, u_1, u_2, ..., u_n, \tau_1, \tau_2, ..., \tau_n \).
   
   In the above rule, \( d_i \) is the premise proposition, \( d \) is the result proposition, \( u_i, \tau_i \) are the CF and threshold of each transition \( i=1,2,...,n \), the weight value on the input arc of each transition is 1, the corresponding FPN model shows in the figure 2.

![Figure 1. The FPN model of rule AND](image1)

![Figure 2. The FPN model of rule OR](image2)

Activation of fuzzy production rules is realized by igniting the transition. For any transition \( t \), if the sum of the products of tagvalues of all input places and weights of their corresponding input arcs is equal to or greater than the threshold of transition \( t \), transition \( t \) is enabled. The definition is as follows.

**Definition 3**

\[ \forall t \in T, \text{ assuming } \forall p_j \in I(t), \sum_{j=1}^{n} M(p_j) \times w_{ij} \geq \tau_j (t) \text{, then transition } t \text{ is enabled, } j=1,2,...,n. \]

The enabled transition could be ignited. When transition \( t \) is ignited, fuzzy reasoning could be done, the tagvalue of the input place of transition \( t \) would not be changed, however it would deliver a new tagvalue \( u \times \sum_{j=1}^{n} M(p_j) \times w_{ij} \) (here \( u \) is the CF of transition \( t \)) to the output place. For Rule AND and Rule OR, their tagvalue results delivered to the output place are...
different. The following definition is used to compute the tagvalue delivered to the output place in Rule OR.

**Definition 4** Assuming place p is the output place of transitions t_i (i=1,2,...,n), then the tagvalue M(p) is the maximal value among n values delivered to place p:

\[ M(p) = \max \left( \sum_{j=1}^{n} M(p_j) \times w_{ij}, \sum_{j=1}^{n} M(p_j) \times w_{2j}, \ldots, \sum_{j=1}^{n} M(p_j) \times w_{nj} \right), p_j \in I(t). \]

Here in order to convert one problem (whether or not a transition is enabled) to another problem (whether or not the independent variable of a continuous function meets certain requirements), and enable the outcome of fuzzy reasoning to be a continuous function (the first-order derivative of which is convenient to be calculated), we use a Sigmoid function to construct a igniting-transition continuous function and a maximum operation continuous function.

### 2.2. Fuzzy Reasoning Function

Assuming y(x) is a Sigmoid function, b is a constant, the expression of y(x) is:

\[ y(x) = \frac{1}{1 + e^{-(x-k)}}. \]

When b is large enough, (1) if \( x > k \) and \( e^{-(x-k)} \approx 0 \), then \( y(x) \approx 1 \); (2) if \( x < k \) and \( e^{-(x-k)} \to \infty \), then \( y(x) \approx 0 \). Obviously, the two-value property of the continuous function y(x) could be regarded as a sign whether or not a transition is enabled.

1. Igniting-Transition Continuous Function

In the definition 3, assuming \( x = \sum_{j=1}^{n} M(p_j) \times w_{ij}, k = \zeta(t) \), then function y(x) has established an enabled-judgment of one transition: When b is large enough, we can know from above, (1) if \( x > k \), y(x)=1, which expresses transition t is enabled; (2) if \( x < k \), y(x)=0, which expresses transition t is not enabled. So we can use the continuous function y(x)×u× \( \sum_{j=1}^{n} M(p_j) \times w_{ij} \) to express whether or not transition t is ignited and the tagvalue output to the place. For the Rule OR, we can also construct a maximum operation continuous function as follows.

2. Maximum Operation Continuous Function

Assuming \( x_1, x_2, x_3 \) are the output values of 3 enabled transitions, use the previous y(x) function, when b is large enough, obviously the derivation procedure as follows is established:

\[ g = \max(x_1, x_2) = \max(1 + e^{-(x_1-k)}), \]
\[ h = \max(x_1, x_2, x_3) = \max(\max(x_1, x_2), x_3) = \max(g, x_3) = \max(1 + e^{-(x_1-k)} + x_3/(1 + e^{-(x_1-k)})). \]

and so on, when there exist many enabled transitions, the corresponding output place p could get a value of the maximum operation continuous function.

### 3. Research Method

Determining the CF, threshold and weight value on the input arc of each transition is crucial for constructing a FPN model. In many practical cases, only relying on experts’ experience is very difficult to obtain those parameters accurately, sometimes even could not be obtained. As a statistical-heuristic combinatorial optimization searching method, the genetic algorithm (which is based on the biological evolution principle) is good at searching an optimal solution in a large, complex and uncertain system and searching for the best individual to meet the needs of that system from a global perspective, so it has been successfully applied to the optimization of some complicated problems in the fields such as neural networks, fuzzy processing, etc [6, 7]. In this paper, the strategies based on an improved genetic algorithm are introduced to realize searching an optimal solution for the parameters of a FPN model.

Assuming each place of p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8 respectively corresponds to a related proposition d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8 in an expert system, the fuzzy production rules exist among them as follows:

\[ R_1 : \text{IF } d_1 \text{ THEN } d_2(u_2, T_2), \]
\[ R_2 : \text{IF } d_1 \text{ or } d_2 \text{ THEN } d_3(u_3, T_1, u_3, T_3), \]
\[ R_3 : \text{IF } d_3 \text{ and } d_4 \text{ and } d_5 \text{ THEN } d_6(w_1, w_2, u_3, T_4), \]
\[ R_4 : \text{IF } d_3 \text{ and } d_7 \text{ THEN } d_8(w_4, w_5, u_3, T_5). \]

According to the above fuzzy production rules, a FPN model could be constructed as shown in Figure 3.
Figure 3. A FPN model case

Figure 4. The FPN model added in a virtual place and a virtual transition

In the FPN model of the Figure 3, \( t_1 \) and \( t_2 \) are the transitions of the 1st layer, \( t_3 \) and \( t_4 \) correspond to the transition of the 2nd layer, \( t_5 \) and \( t_6 \) correspond to the transition of the 3rd layer and \( t_7 \) and \( t_8 \) correspond to the transition of the 4th layer. Because \( t_1 \) and \( t_3 \) do not lie in the same layer, but they all correspond to the same output place \( p_3 \), in order to construct a better hierarchical model and conduct to lighting the transitions layer by layer in the process of fuzzy reasoning, we introduce a virtual place \( p_9 \) (represented by a hollow circle) and a virtual transition \( t_6 \) (represented by a small hollow rectangle) between \( t_1 \) and \( p_3 \) [1], at the same time cancel the output arc from \( t_1 \) directly pointing to \( p_3 \), and replace it with one from \( t_1 \) pointing to the virtual place \( p_9 \), then add an input arc from \( p_9 \) pointing to the virtual transition \( t_6 \) and an output arc from \( t_6 \) pointing to \( p_3 \). The FPN model added in a virtual place and a virtual transition is shown in Figure 4. The hierarchy in the Figure is as follows: \( t_1 \) and \( t_2 \) are the transitions of the 1st layer, \( t_3 \) and \( t_6 \) are the transitions of the 2nd layer, \( t_4 \) and \( t_5 \) still correspond to the transition of the 3rd and the 4th layer respectively. The virtual place and the virtual transition only play an intermediate transitional role in the FPN model, adding them has no effect on the rule base system, the threshold of the virtual transition is 0, and the CF is 1.

3.1. Fuzzy Reasoning

Using the fuzzy reasoning function and hierarchy as constructed above, igniting the model layer by layer, and the reasoning order is as follows:

1. Firstly igniting the transition \( t_1 \) and \( t_2 \) in the 1st layer,
   \[
   M(p_{n}) = M(p_{n}) \times u_{1}/(1 + e^{-b(M(p_{n}) - r_{1})})
   \]
   \[
   M(p_{n}) = M(p_{n}) \times u_{2}/(1 + e^{-b(M(p_{n}) - r_{2})})
   \]
2. then igniting the transition \( t_3 \) and \( t_6 \) in the 2nd layer,
   \[
   x_{1} = M(p_{n}) \times x_{2} = M(p_{n}) \times u_{3}/(1 + e^{-b(M(p_{n}) - r_{3})})
   \]
   \[
   M(p_{n}) = \max(x_{1}, x_{2}) = x_{1}/(1 + e^{-b(x_{1} - x_{2})}) + x_{2}/(1 + e^{-b(x_{2} - x_{1})})
   \]
3. next igniting the transition \( t_4 \) in the 3rd layer,
   \[
   x_{3} = M(p_{n}) \times w_{1} + M(p_{n}) \times w_{2} + M(p_{n}) \times w_{3}
   \]
   \[
   M(p_{n}) = x_{3} \times u_{4}/(1 + e^{-b(x_{3} - r_{4})})
   \]
4. finally igniting the transition \( t_5 \) in the 4th layer,
   \[
   x_{4} = M(p_{n}) \times w_{4} + M(p_{n}) \times w_{5}
   \]
   \[
   M(p_{n}) = x_{4} \times u_{5}/(1 + e^{-b(x_{4} - r_{5})})
   \]
3.2. Learning and Training in the FPN Model

After determining the reasoning process of the FPN model, we could commence training the above FPN model by using an improved genetic algorithm [6].

(1) Coding
In order to speed up the convergence of the GA and guarantee the necessary accuracy, we adopt a decimal coding scheme for weight, CF and threshold of each transition.

(2) Fitness Function
A square sum of the difference between a batch of sample output and desired output of the network is adopted for the fitness value, the target is to obtain a set of parameters which has the minimum fitness value. The function definition is as follows:

$$E_i = 2^{-1} \sum_{j=1}^{r_1} (M_{i,j}(p_k) - M^*_i(p_k))^2$$

In the expression, $i=1,2,...,r_1$ is the population number of code for parameters, $j=1,2,...,r_2$ is the sample number of initial input, $M_{i,j}(p_k)$ and $M^*_i(p_k)$ respectively expresses the actual tagvalue and the desired tagvalue of the $i$th individual under the $j$th input sample.

(3) Genetic Manipulation
In order to maintain the diversity of solutions produced in the process of group crossing, and prevent premature phenomenon, we improve the basic genetic manipulation as follows:

Step 1. During the process of duplication, in all of the N individuals, the superior ones (the fitness value of which are less than the average number) would be holded, the inferior ones (the fitness value of which are greater than the average number) would be eliminated one quarter of N, and the eliminated individuals would be supplemented by means of generating random numbers.

Step 2. In order to ensure that the global optimal solution could be found, the optimum in the current iteration would be directly evolved into the next iteration.

Step 3. During the process of crossing manipulation, the optimum in the last iteration would be placed into the current iteration as the first individual, one or two decimal digit(s) after the decimal point from the 2nd to (N/2)th individual would be exchanged with the digit(s) in the same position of the individuals from the (N-1)th to (N/2)+1)th, so the new (N/2)-1 individuals are formed; at the same time, one or two decimal digit(s) after the decimal point from the (N/2)+1)th to Nth individual would be exchanged with the digit(s) in the same position of the individuals from the 1st to (N/2)th, then the rest (N/2) individuals are formed.

Step 4. The variation method used in this paper is to change one natural number of one individual into another natural number, the proportion of mutation would be upgrading according to the increase of evolution iterations, the purpose of which is to increase the randomness and then to increase the diversity. The mutation rate would be gradually changed from 0.05 to 0.1 during the whole process of searching.

According to the above method, we could draw an algorithm flow chart of searching the optimal parameters in the FPN model by using an improved genetic algorithm as shown in Figure 5.

![Figure 5. The algorithm flowchart](image-url)
4. Results and Analysis

Assuming the ideal parameters of the FPN model in Figure 4 are as follows: $w_1=0.2$, $w_2=0.5$, $w_3=0.3$, $w_4=0.4$, $w_5=0.6$, $u_1=0.7$, $u_2=0.9$, $u_3=0.6$, $u_4=0.8$, $u_5=0.7$, $\tau_1=0.3$, $\tau_2=0.4$, $\tau_3=0.2$, $\tau_4=0.5$, $\tau_5=0.4$, the constant $b$ in the fuzzy reasoning function is 5000, the population number $r_1$ is 100, the sample number $r_2$ is 30, then according to the Standard Genetic Algorithm (SGA), the trained parameters after 500 iterations are $w_s$, $u_s$, $\tau_s$, which are shown in Table 1, and according to the Improved Genetic Algorithm (IGA), the trained parameters after 200 iterations are $w_i$, $u_i$, $\tau_i$, which are also comparatively shown in Table 1. MSE is the sum of Mean Square Error (Unit: $1.0 \times 10^{-3}$). Obviously, the parameters obtained by using the IGA are better than those by using the SGA.

In order to test its generalizing capability, we randomly select 6 sets of input data which are not in the sample, and apply fuzzy reasoning to the FPN model which is trained by the IGA. The result of fuzzy reasoning is shown in the column of Actual Output in the Table 2, which reflects the degree value of membership of the result proposition $M(p_i)$, the mean value of total squared error on the output is 0.001461, the result is satisfactory.

<table>
<thead>
<tr>
<th>PRM</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>MSE($10^{-3}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_s$</td>
<td>0.21</td>
<td>0.51</td>
<td>0.29</td>
<td>0.45</td>
<td>0.69</td>
<td>2.18</td>
</tr>
<tr>
<td>$w_i$</td>
<td>0.21</td>
<td>0.50</td>
<td>0.32</td>
<td>0.40</td>
<td>0.56</td>
<td>0.42</td>
</tr>
<tr>
<td>$u_s$</td>
<td>0.74</td>
<td>0.84</td>
<td>0.70</td>
<td>0.79</td>
<td>0.66</td>
<td>3.38</td>
</tr>
<tr>
<td>$u_i$</td>
<td>0.65</td>
<td>0.89</td>
<td>0.62</td>
<td>0.81</td>
<td>0.64</td>
<td>1.34</td>
</tr>
<tr>
<td>$\tau_s$</td>
<td>0.31</td>
<td>0.39</td>
<td>0.35</td>
<td>0.40</td>
<td>0.27</td>
<td>9.92</td>
</tr>
<tr>
<td>$\tau_i$</td>
<td>0.21</td>
<td>0.46</td>
<td>0.22</td>
<td>0.44</td>
<td>0.49</td>
<td>4.76</td>
</tr>
</tbody>
</table>

5. Conclusion and Prospect

In this paper, an improved genetic algorithm is applied to search the optimal solution of all parameters in the FPN model, which would make the FPN model possess the learning capability like the neural network, and the obtained parameters are very close to the ideal parameters. However, the genetic algorithm belongs to a kind of global-searching algorithm, and too many optimal parameters need to be searched, the trained parameters are not all identical to the ideal parameters, particularly there exist some differences for one or two thresholds and CFs. But if most of parameters have been determined and there exist only a few parameters to be searched, or the scope of initial input could be approximately determined, the strategies of searching the optimal solution proposed in this paper are very accurate and applicable. In the future research work, some characteristics of the neural network (especially for the capability of BP network being good at local-searching) combined with the global-searching characteristic of the genetic algorithm should be taken into consideration, so as to further enhance the accuracy of optimizing the parameters in the FPN model.
Acknowledgements

The authors wish to acknowledge the funding support from the Natural Science Foundation (NSF) Project (01 JJY2061) and the Scientific Research Foundation (SRF) Project (01 C306) of Hunan Province Education Office.

References


