Computing and Informatics, Vol. 32, 2013, 1001–??, V 2014-Aug-14

FUZZY MULTI-STATE SYSTEM RELIABILITY ANALYSIS USING WEIGHTED FUZZY PETRI NETS

Xinju Zhang[1], Shuzhen Yao[2]

[1] *NTT, Musashino-shi, 180-8585 Japan*
[2] *IEICE, Minato-ku, Tokyo, 105-0011 Japan*
e-mail: `juzi200501@163.com`

Abstract. Currently available methods for evaluating the reliability of multi-state systems are not suitable for fuzzy multi-state systems because they cannot process a fuzzy multi-state system with fuzzy reliability and because they lack flexibility. To remedy this limitation in fuzzy multi-state system reliability assessment, a weighted fuzzy Petri net reliability evaluation method is proposed for fuzzy multi-state systems. First, a weighted fuzzy Petri net model of the fuzzy multi-state system is established in which the transitions have fuzzy weights in [0,1] and in which the input and output of each transition are node states. Second, the weights of the fuzzy status of the weighted fuzzy Petri nets are calculated based on an improved particle algorithm utilizing a two-stage optimization strategy to make the weights more accurate. Finally, the transient-state reliability is evaluated based on their weights with a fuzzy universal generating function. MATLAB simulations show that the proposed reliability evaluation method based on the weights of the weighted fuzzy Petri nets is more accurate than previous methods and provides theoretical guidance for system reliability evaluations.

Keywords: WFPN, Reliability, Weight, FUGF

Mathematics Subject Classification 2010: AB-XYZ

1 INTRODUCTION

A fuzzy multi-state system is a system that may be in more than one state at a certain moment, and because fuzzy multi-state systems predominate in actual industrial production processes, reliability evaluations are essential. However, most current research is aimed at evaluating the reliability of multi-state components or of systems under steady-state conditions, reflecting the stable performance distribution of components or system states under infinite running time conditions. In real projects, we often pay more attention to the dynamic reliability behavior of the system, which can directly reflect the status and residual life of the current components and systems, to effectively prevent accidental failures and reduce the potential risks of system failures. Due to limitations of time, money, or state combinations, a system's dynamic transformation has a certain ambiguity. Therefore, in this paper, the dynamic performance of a fuzzy-state system and the theory of reliability evaluation are further studied and simulated using MATLAB.

Fuzzy system reliability theory combines fuzzy mathematics and system reliability to address the fuzzy phenomena involved in system reliability. It is a useful supplement to conventional reliability and is the dominant method for addressing fuzzy uncertainty. Therefore, multi-state theory and fuzzy theory are critical in addressing the reliability of fuzzy multi-state system, which has become a focus of scholarly research.

The fuzzy Petri net model is based on fuzzy production rules but lacks the ability to learn. One of the parameters is usually based on expert experience, but accurate information could be difficult or even impossible to obtain. However, fuzzy reliability theory for a fuzzy multi-state system has been seldom discussed in previous research. For such systems, the state probabilities of a component and/or the state performances of a component can be estimated as fuzzy values; therefore, the transfer rate of every state is also a fuzzy value. Moreover, in the fuzzy multi-state system model, the transfer rates of system failure and success may possibly include some system states. Therefore, in this paper, a weighted fuzzy Petri net (WFPN) model of a fuzzy multi-state system is first established, and subsequently, the transfer rate is assumed as the weight of the WFPNs model. Second, the weight of every fuzzy state is estimated using an improved particle algorithm, and the probability is changed using the weights. Finally, the expected transient reliability performance is estimated based on the fuzzy universal generating function (FUGF).

2 RELATED WORK

There are few methods for assessing the reliability of a fuzzy multi-state system. Some of these methods are based on adaptive fuzzy stochastic Petri nets and on node importance and dependence. The first multi-state reliability studies were performed in the 1970s [1]. The methods for analyzing the reliability of multi-state systems using Petri nets primarily include structural function methods, generation function methods, and Monte Carlo simulations [2, 3].

Petri nets have increasingly been employed in modeling. Petri nets are bipartite-directed graphs that include an initial mark. The nets comprise two types of nodes, places and transitions, which are connected by arcs. Arcs may connect a place to a transition or a transition to a place. Graphically, transitions are usually depicted as bars, and places are depicted as circles [4]. It is also often important to model uncertainty, which has been accommodated by further extensions to Petri nets. These extensions capture and reflect fuzzy knowledge [5]. The use of fuzzy Petri nets in the reduction of chaos was first proposed in 1992 (Bechtold, 1991) [6]. One key issue in constructing a process model based on fuzzy Petri nets is the periodic retrogressions that develop in the process.

Fuzzy Petri nets were first used to estimate the traits of multi-state systems in 1989, when Valette proposed the construction of libraries denoted by fuzzy sets. In this system, each transition is given a fuzzy time [7]. Looney proposed that fuzzy Petri nets are capable of learning and that this ability could be used to adjust thresholds and limit the reliability analysis of existing systems [8]. The constant rate of change in stochastic Petri nets can be extended to fuzzy numbers to achieve fuzzy-number-based repairable system reliability estimations [9]. In a previous study, a BP(Back Propagation) neural network algorithm was introduced to FPNs [10]. An error back-propagation algorithm using a ladder degree approach to fuzzy production was developed to train parameters. Fuzzy stochastic Petri nets will be introduced in the following section. The report "Characterizing Liveness Monotonicity for Weighted Petri Nets in Terms of Siphon-Based Properties" (Li Jiao and To-Yat Cheung, 2003) [11] introduced three new siphon-based characterizations for these properties. In addition, the problem of distributed causal model-based diagnosis in interacting behavioral Petri nets (BPNs) has also been discussed [12].

3 WEIGHT OF WFPNS LEARNING

Table 1. Nomenclature

j	Component index for fuzzy multi-state system.
k_j	Number states for component j
$\tilde{g}_J = \{g_{1i_1} \ldots g_{jk_j}\}$	Performance set for component j
$\tilde{p}_J = \{p_{1i_1} \ldots p_{jk_j}\}$	Probability set for component j
ϕ	Fuzzy multi-state system structure function
i	State index for a fuzzy multi-state system
ω_{ij}	The weight from the state i to transition j
ω'_{kj}	The weight from transition k to the state j. To simplify the calculation, it is assumed to be negligible in the simulation.
ω_{ji_jk}	The weight from the state i_j of component j to the transition k, sometimes simply denoted by ω_{jk}.
$\tilde{p}_i$	Probability of the fuzzy multi-state system being in state i represented by a fuzzy set
$\tilde{g}_i$	Performance rate of the fuzzy multi-state system being in state i represented by a fuzzy set
$f(x)$	Membership function of the fuzzy arc weight
x	The variable of the fuzzy arc weight

$\mu_{g_{ji_j}}$	Membership function of g_{ji_j}
$\mu_{\tilde{g}_i}(g_i)$	Expresses the membership functions of $\mu_{g_{ji_j}}$.
$U(z)$	Fuzzy universal generating function for a fuzzy multi-state system
W	Weight vector

3.1 Weighted Fuzzy Petri nets

Weighted fuzzy Petri nets (WFPNs) are defined as a ten-dimensional group [10]:

$$S_{WFPN} = \langle P, D, T, I, O, \alpha, T_h, W, \beta, \tau \rangle \,.$$

$P = \{P_1, P_1 \ldots P_n\}$ is a finite set of places that are used to define the fuzzy multi-state system component.

$D = \left\{d_{ij}, d'_{ij}\right\}$ is a finite set of the fuzzy state, whereby every state is represented as a token. The state d_{ij} is from the place P_i to the transition t_j, and the state d'_{ij} is from the transition t_j to the place P_i.

$T = \{t_1, t_2 \ldots, t_m\}$ is a finite set of transitions, which are used to define the transition of the state.

$I : P \to T$ is the input function, representing a mapping from transitions to their input places and denoted as $^*t$;

$O : T \to P$ is the output function, representing a mapping from transitions to their output places and denoted as t^*;

$\alpha(d_{ij}, t_k)$ and $\alpha(t_k, d'_{ij})$ in [0, 1] are the fuzzy numbers that represent the probability of every fuzzy state that are in a place and the likelihood of transitions. $\beta : P \to D$ such that when $P_i \in P$ has a special $token_{ij}$, then $\beta(token_{ij}, P_i) = d_{ij}$ or $\beta(token_{ij}, P_i) = d'_{ij}$. When t_k fires, the premise condition to transition of the state d_{ij} is $\alpha(d_{ij}, t_k)$.

$Th : Th \to [0, 1]$ is the threshold of the transition $Th = \{\lambda_1, \lambda_2, \ldots, \lambda_m\}$.

$$\begin{aligned} W = \{&\omega_{11} \ldots \omega_{1m}; \omega_{21} \ldots \omega_{2m}; \ldots \omega_{n1} \ldots \omega_{nm}; \\ &\omega'_{11} \ldots \omega'_{1r}; \omega_{21} \ldots \omega'_{2r}; \ldots \omega'_{m1} \ldots \omega'_{mr}\}, \end{aligned}$$

where $\omega_{ij} \in [0, 1]$, which is a fuzzy number, denotes the weight from a place to a transition, namely, the different degrees of support from one place to different transitions. The membership function of the fuzzy arc weight could be assumed to follow a half normal distribution [11]

$$f(x) = f(\omega_{ij}) = \left\{ \begin{array}{ll} 1 & \omega_{ij} \le a \\ 1 - e^{-k(\omega_{ij} - a)^2} & \omega_{ij} > ak > 0 \end{array} \right\} \tag{1}$$

where a and k are constants.

$\tau = \{\tau_1, \tau_2 \ldots \tau_n\}$ is the fuzzy set of the transition rate, which denotes the average number of times that an event takes place under the given conditions per unit of time. The initiator is always a positive real number.

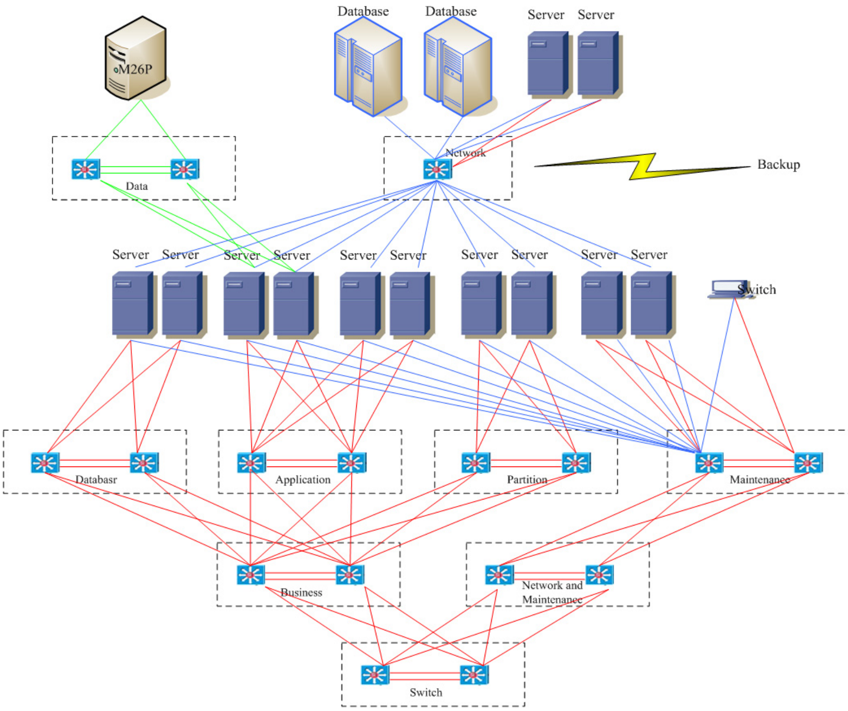

Fig. 1. Bank switch model

Definition 1. $\forall t_k \in T, \sum_{i=1}^{n} \alpha(d_{ij}, t_k) - \omega_{ij} \geq \lambda_k$, then the transition $t_k (k = 1, 2 \ldots m)$ is enabled.

Definition 2. The firing rules of transitions. When the transition t_k is fired, the sum of the probabilities of every input state minus the corresponding support degree is the support degree to the next component state.

$$\alpha(t_k, d'_{ij}) = \sum_{i=1}^{n} \alpha(d_{ij}, t_k) - \omega_{ik} - \omega'_{kj} \quad (j = 1, 2 \ldots n, k = 1, 2 \ldots m) \tag{2}$$

Definition 3. When the place P_i is the output place of many transitions t_k ($k = 1, 2 \ldots m$) and when the transitions are fired, the support degree to the next component state is

$$\max\left(\sum_{i=1}^{n} \alpha(d_{ij}, t_k) - \omega_{ik} - \omega'_{kj}\right) \tag{3}$$

For example, for a bank payment system, when an emergent event occurs, the main system must switch to a backup system to ensure stability, and the actual business data is transmitted to validate the backup system's availability. In this paper, the system switches are used to verify the reliability analysis. In the process of switching, each business system, encryption system and certification system is required to switch to the backup system, and the backup system topology is adjusted at the same time. Because the switching process is affected by the environment and by other uncertainty factors, the switching subsystem will present multiple fuzzy states, such as failure and success. The bank switch model is as follows:

For the bank switch model, because every subsystem is regarded as one component that could be presented as a place and because every place could have many tokens, which express each state of every subsystem, the probability of each state is a fuzzy number. P_0 is the system master; when there is a token, which denotes that the main system has failed, the switch is enabled. P_1 is the business system, P_i is the encryption system, P_n is the authentication system, P_1' is the backup business system, P_1' is the backup encryption system, P_{n-1}' is the timing system of the backup system, P_n' is the backup authentication system, the transition $t_j(j = 1, 2 \dots, m)$ denotes the switching process, and $\omega_{ij}, \omega_{ij}'$ denote the transition weights, which indicate the transfer rate between each state. The weighted fuzzy Petri nets of the multi-state switching system are as follows:

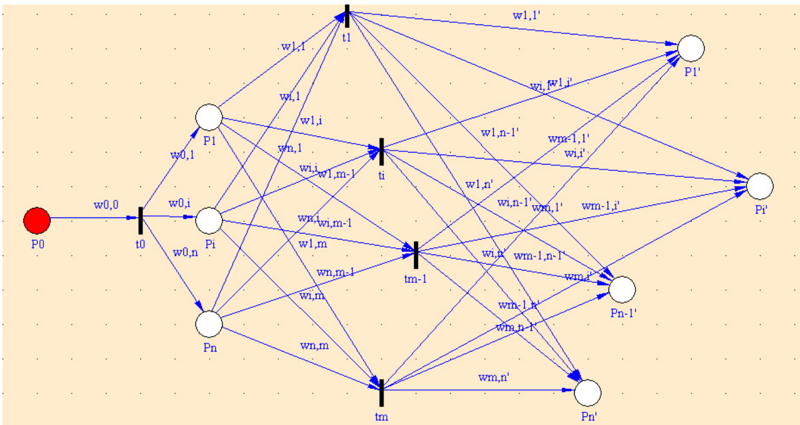

Fig. 2. WFPNs of the bank switching model

3.2 Weight learning with improved particle algorithm

The performance of the fuzzy multi-state system and the system state probability distribution or transfer rate are described using fuzzy numbers. In this case, according to Ding [13], it is supposed that the system element j can have k_j different states. The fuzzy probability $\tilde{p}_i$ of state i of the fuzzy multi-state system is calculated as

$$\tilde{p}_i = \left\{ p_i, \mu_{\tilde{p}_i}(p_i) | p_i = \prod_j^n p_{ji_j} \right\}, \tag{4}$$

where $\mu_{\tilde{p}_i}(p_i) = \sup_{p_i=\prod_{j}^{n} p_{ji_j}} \min\left\{\mu_{p_{1i_1}} \ldots \mu_{p_{ji_j}}\right\}$.

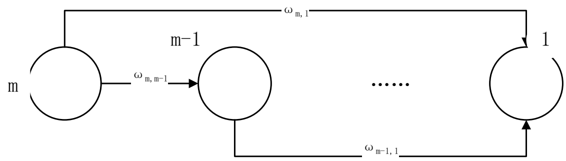

Fig. 3. State space

$\tilde{g}_i$ is the performance of system state i, which could be evaluated as

$$\tilde{g}_i = \left\{g_i, \mu_{\tilde{g}_i}(g_i) | g_i = \phi(g_{1i_1} \ldots g_{ji_j} \ldots g_{ni_n})\right\} \tag{5}$$

where $\mu_{\tilde{g}_i}(g_i) = \sup_{\phi(g_{1i_1} \ldots g_{ji_j} \ldots g_{ni_n})} \min\{\mu_{g_{1i_1}} \ldots \mu_{g_{ji_j}}\}$, and $\phi(g_{1i_1} \ldots g_{ji_j} \ldots g_{ni_n})$ is the structure function.

j, i_j indicate the j-th place and the i_j-th status. According to the fuzzy universal generating function proposed by Ding and Lisnianski [13], the transient-state distribution can be expressed as a polynomial; for the sake of simplicity, ω'_{ji_jk} is assumed to be negligible.

The fuzzy Probability Distribution (PD) of the place j can be represented using the following form:

$$\mu(z) = \sum_{i_j=1}^{k_j} \tilde{p}_{ji_j} z^{\tilde{g}_{ji_j}} \tag{6}$$

The fuzzy PD of a fuzzy multi-state system is used over the z-transform fuzzy representations of n system components [14, 15].

$$U(z) = \sum_{i_1=1}^{k_1} \ldots \sum_{i_2=1}^{k_2} \sum_{i_n=1}^{k_n} \tilde{p}_i z^{\tilde{g}_i} \tag{7}$$

According to Definition 2 and Definition 3, in the FPN, the fuzzy probability $\tilde{p}_i$ of the state i from the state k can be calculated as

$$\tilde{p}_i = \max(\sum_{i=1}^{n} \alpha(d_{kj_i}, t_k) - \omega_{kj_i}) \tag{8}$$

In the WFPNs model, ω_{kji_j} denotes the weight from a place to a transition, namely, different degrees of support from one place to a different transition. According to formula (7), the most important step is estimating the instantaneous probability and transfer rate between states. In a fuzzy multi-state system, the state space of a node m can be represented by the following figure:

In fig. 3, the state transition rate between any two states m and $m-1$ is represented by the fuzzy number $\omega_{m,m-1}$. Because the transfer rate is fuzzy, the probability distribution of any node is also fuzzy.

In the WFPNs model of a fuzzy multi-state system, as the number of nodes increases, the feasible solution space of the reliability optimization problem grows exponentially, which presents new challenges for the optimization algorithm. In recent years, a variety of computational intelligence methods have been developed to provide a new approach for determining the weights in multi-state system reliability optimization. P.A.V. de Miranda [16] introduced a framework for synergistic arc-weight estimation, where the user draws markers inside each object (including the background) and where arc weights are estimated from image attributes and object information. Levitin and Lisnianski [14] [15] used the advantage of global convergence in particle algorithms to solve multi-state system reliability optimization problems. Tian et al. [17] solved the multi-state system redundancy-reliability optimization problem using a genetic algorithm. Levitin [18] solved multi-state system reliability optimization and multi-objective optimization with a single-objective particle swarm algorithm. Meziane et al. [19] and Sharma et al. [20] applied the ant colony algorithm to a constrained optimization of multi-state system reliability.

Victor and Shen developed a reinforcement learning algorithm for the high-level FPN models to simultaneously perform structure and parameter learning [21]. Since the proposal of the particle swarm algorithm in 1995, it has drawn widespread attention in the field of numerical optimization.

At present, the main challenge is to avoid premature convergence while obtaining high convergence speed. Most efforts toward avoiding premature convergence have focused on forcing the algorithm to jump out of local minima. To increase the convergence speed, efforts have been made toward choosing optimal algorithm parameters. In this paper, we solve the problem of premature convergence based on control population information dissemination and the propagation velocity, increase the convergence rate using global and local search capabilities, and propose a comprehensive evaluation model for WFPNs.

Among these possibilities for fuzzy weight assessment, the improved particle swarm algorithm is adopted for a two-stage optimization of the fuzzy weights. First, in the particle swarm algorithm, fuzzy weighting parameters are introduced in the optimization process; subsequently, a two-stage parameter optimization process that does not rely on empirical data and that has no strict requirements on the initial input is proposed. The main concept of the algorithm is as follows:

1. The parameters are optimized to avoid slow convergence and to avoid falling into local minima. The input samples for the WFPNs model of the fuzzy multi-state system do not immediately take effect, but after all samples in an entire cycle are sequentially input, the sum of the errors is calculated, and the parameters are modified together.
2. To adjust the parameters, particle swarm optimization is used. The adjustment

formula is as follows:

$$\begin{aligned} v_{id}^{t+1} &= \alpha v_{id}^{t} + c_1\gamma_1^t(p_{ld}^t - x_{id}^t) + c_2\gamma_2^t(p_{gd}^t - x_{id}^t) \\ x_{id}^{t+1} &= x_{id}^t + v_{id}^{t+1} \end{aligned} \tag{9}$$

The weight ω_{ij} is supposed as the particle, where l denotes the best index of the particles in each area and g denotes the best overall index of the particles. p_{ld}, p_{gd} denote the optimal value of each area and the optimal value of all particles, respectively.
α is used to maintain the original speed factor, that is, the inertial weight.
c_1, c_2 denote the best weight of each area and the best weight of the group particles, respectively.
γ_1, γ_2 are uniformly distributed random numbers between 0 and 1.
v_{id}^{t+1} is the fitness function.
W is the objective value.

3. With the improved particle swarm optimization algorithm, the WFPNs become multi-valued logic Petri nets, and the continuous function of the transitions is as follows:

$$\max\left(\sum_{i=1}^{n}\alpha(d_{ij}, t_k) - \omega_{ik} - \omega'_{kj}\right) \tag{10}$$

Now, the improved algorithm is applied to the fuzzy weight optimization.

Algorithm 1:
Input: initial weight vector ω_0, Iterations $d = 0$, and allowable error ε;
Output: weight vector W.

1. A WFPNs model of a fuzzy multi-state system is established.
2. Transitions are fired by the firing rules, and the true degree of probability is calculated as follows: $\max(\sum_{i=1}^{n}\alpha(d_{ij}, t_k) - \omega_{ik} - \omega'_{kj})$.
3. The weight vector of the WFPNs models is adjusted via learning according to the true degree of credibility, using the two-stage parameter optimization strategy of the improved particle algorithm.
4. The first stage of the improved particle algorithm:
   (1) Iterations $d = 0$, allowable error ε, initial weight vector $W_0 = \{\omega_0\}$.
   (2) To increase the particle convergence speed, the weight is applied with decimal encoding.
   (3) The final confidence error vector is summed.

$$E(\omega_{ij}) = \frac{1}{2}\sum_{z=1}^{r}\sum_{i=1}^{a}(\theta_i^z - (\theta_i^z)^l)^2,$$

where $\theta_i^z, (\theta_i^z)^l$ represents the actual value and the desired value of the z-th particle datum of the terminal place p_i and which is calculated using formula (3); a represents the number of the terminal place; and r represents the batch number of the sample data. If $|E(\omega_{ij})| < \varepsilon$, go to (6); otherwise, go to (4).

(4) The weight vector is adjusted using the particle swarm algorithm, and the position and velocity are calculated using equation (10).
(5) If $|E(\omega_{ij})| < |E(\omega_{ij+1})|$, then $d = d+1$, and go to (3). If $|E(\omega_{ij})| = |E(\omega_{ij+1})|$, go to (4).
(6) The adjustment is completed, and the weight vector is updated such that the new weight parameter vector is $W = \{\omega_{ij}\}(i = 1 \dots n, j = 1 \dots m)$.

After multiple iterations, the population is basically stable, but the weight vector of the WFPNs model requires further improvement. In the second stage, the trained weight vector is precisely adjusted again using the fitness transformation function.

The second stage of the improved particle algorithm is as follows:

In the first stage, p_{gd} is directly determined from the most successful particles, but its search direction tends to be controlled by the few best super-particles, which do not necessarily guide the model in the direction of the global optimum; thus, premature convergence can easily occur. To avoid this problem, we used a transformation function to ensure that the probability of p_{gd} is inversely proportional to the target value of its subordinate function and that, after the transformation, the adaptive value is not less than zero. Because there are numerous particles in the improved algorithm, and because the factor is complex, the fitness of the transformation function in the optimization stage [8] is selected as follows:

$$FS(f(x)) = \frac{m}{m + f(x) - GM} \tag{11}$$

where GM is the extremum value of the weight subordinate function $f(x)$; m reflects the normal transformation scale; and $f(x)$ is the weight subordinate function. To determine a particle's fitness, the degree of influence of each particle is calculated.

$$SI(i) = \frac{FS(f(x_i))}{\sum\limits_{i=1}^{gd} FS(f(x_i))} \tag{12}$$

Combined with the degree of influence of each particle, the current value p_{gd} of the best particle in the standard particle algorithm is considered using the weight ps of the comprehensive weight model.

$$ps = \sum_{i=1}^{gd} SI(i) \times x_i \tag{13}$$

Thus, the speed/position update formula of the updated particle algorithm is as follows:

$$\begin{aligned} v_{id}^{t+1} &= \alpha v_{id}^{t} + c_1 \gamma_1^t (p_{ld}^t - x_{id}^t) + c_2 \gamma_2^t (ps^t - x_{id}^t) \\ x_{id}^{t+1} &= x_{id}^t + v_{id}^{t+1} \end{aligned} \tag{14}$$

1. Iterations $d = 0$, allowable error ε, the first stage weight vector W.

2. Calculate the total error $E(\omega_{ij}) = \frac{1}{2}\sum_{z=1}^{r}\sum_{i=1}^{a}(\theta_i^z - (\theta_i^z)^l)^2$. If $|E(\omega_{ij})| < \varepsilon$, go to (5); otherwise, go to (3).
3. Adjust the weight using formula (15).
4. If $|E(\omega_{ij})| < |E(\omega_{ij+1})|$, then $d = d+1$, and go to (2). If $|E(\omega_{ij})| = |E(\omega_{ij+1})|$, go to (3).
5. The adjustment is completed, and the weight vector is updated such that the new weight parameter vector is $W = \omega_{ij}(i = 1 \dots n, j = 1 \dots m)$.
6. The improved particle algorithm is completed, and the new weight is W.

4 RELIABILITY ASSESSMENT WITH WEIGHT OF WFPNS

The weights of the WFPNs can be obtained from the above algorithm, and for fuzzy multi-state WFPNs, the reliability can be expressed based on the expected performance. The reliability assessment uses matrix operations.

The fuzzy state performance of the system is mapped from the fuzzy state performance of the components by formula (8), which can produce the fuzzy general generating function of the fuzzy multi-state system. The function is defined by the system structure and the properties of all of the components. Thus, the fuzzy state performance of any state in the fuzzy multi-state system can be determined by the function relating all of the component state performances. The performance rates and probabilities of the components are also assumed as the triangular fuzzy numbers represented as the triplet (a, b, c), which is one of the most important classes of fuzzy numbers and is used in many practical situations [23].

For parallel subsystems, the structure function is the sum of component performances. According to (6) and according to the fuzzy arithmetic operations of the triangular fuzzy numbers [22], the subsystem performance $\tilde{g}_i$ can be obtained as

$$\begin{aligned} \tilde{g}_i = \tilde{\varphi}_P(\tilde{g}_{1i_1}, \dots, \tilde{g}_{ji_j}, \dots, \tilde{g}_{ni_n}) &= (\sum_{j=1}^{n} a_{ji_j}, \sum_{j=1}^{n} b_{ji_j}, \sum_{j=1}^{n} c_{ji_j}) \\ &= (\sum_{j=1}^{n}(\alpha(d_{ij_i}, t_k) - \omega_{kj_i})_a, \sum_{j=1}^{n}(\alpha(d_{ij_i}, t_k) - \omega_{kj_i})_b, \sum_{j=1}^{n}(\alpha(d_{ij_i}, t_k) - \omega_{kj_i})_c) \end{aligned} \tag{15}$$

where $\tilde{\varphi}_P$ is the fuzzy parallel operator and the component $\tilde{g}_{ji_j}$ is represented as the triplet $(a_{ji_j}, b_{ji_j}, c_{ji_j})$. According to (5) and according to the fuzzy arithmetic operations of the triangular fuzzy numbers [22], the subsystem probability $\tilde{p}_i$ can be obtained as

$$\begin{aligned} \tilde{p}_i &= (\prod_{j=1}^{n} a'_{ji_j}, \prod_{j=1}^{n} b'_{ji_j}, \prod_{j=1}^{n} c'_{ji_j},) \\ &= (\prod_{j=1}^{n}(\alpha(d_{ij_i}, t_k) - \omega_{kj_i})'_a, \prod_{j=1}^{n}(\alpha(d_{ij_i}, t_k) - \omega_{kj_i})'_b, \prod_{j=1}^{n}(\alpha(d_{ij_i}, t_k) - \omega_{kj_i})'_c) \end{aligned} \tag{16}$$

where the component p_{ji_j} is represented as $(a'_{ji_j}, b'_{ji_j}, c'_{ji_j})$.

For a series subsystem, if it has more than two components, $\tilde{\varphi}_s$ can be approximated by a triangular fuzzy number:

$$\begin{aligned}\tilde{\varphi}_s &= (\tilde{g}_{1i_1}, \tilde{g}_{2i_2}) = (\min(a_{1i_1}, a_{1i2}), \min(b_{1i_1}, b_{1i2}), \min(c_{1i_1}, c_{1i2}))\\ &= (\min(\alpha(d_{ij_i}, t_k) - \omega_{kj_i})_{a_1}, (\alpha(d_{ij_i}, t_k) - \omega_{kj_i})_{a_2}),\\ &\quad \min(\alpha(d_{ij_i}, t_k) - \omega_{kj_i})_{b_1}, (\alpha(d_{ij_i}, t_k) - \omega_{kj_i})_{b_2}),\\ &\quad \min(\alpha(d_{ij_i}, t_k) - \omega_{kj_i})_{c_1}, (\alpha(d_{ij_i}, t_k) - \omega_{kj_i})_{c_2}).\end{aligned} \tag{17}$$

Thus, in the case of incomplete information, the state reliability, which is based on the reliability analysis using WFPNs and which can also be used to assess a possible fuzzy multi-state design under uncertainties over if it can meet the desired reliability requirements, can be sorted.

Briefly, in [10], a BP network reliability approach based on a fuzzy reasoning algorithm and a learning algorithm for FSPNs was proposed, and the details of that approach and the proposed algorithm are compared below:

1. The improved particle algorithm of WFPNs is proposed in this paper using a one-way flow of information, and the updating process is to follow the process of searching for the optimal solutions; the PSO also has a memory. In addition, in the improved particle algorithm, the particle position and velocity are modeled and a set of significant evolution equations is given, which is the key aspect of the algorithm and can be used to demonstrate its many advantageous characteristics.
2. The improved particle algorithm has a strong global search ability and uses random initialization and an adaptive value evaluation system to conduct a random search according to the fitness value. Therefore, the search method has global characteristics, a simple algorithm, and a fast convergence speed. However, it has a weak local search ability compared to the learning algorithm of the BP network.
3. The improved particle algorithm disturbs each particle using an adaptive function, improving the diversity of the particle swarm. As the number of iterations increases, the magnitude of the perturbation decreases, which guarantees convergence. In addition, to prevent the possibility of the particle swarm quickly converging to a local extremum, we can adjust the rand parameter value to be negative, thus increasing the diversity of the particle swarm. The experimental results demonstrate the effectiveness of the proposed method, and the learning algorithm of the BP network does not have the ability.

5 SIMULATION

Take fig. 1 as an example for the simulation. First, the weight vector of the WFPNs is determined using the improved particle algorithm with iterations $d = 0$, allowable error $\varepsilon = 0.009$, population size 300, and initial weight vector ω_{ji_jk} as a random number between 0 and 1; the transition threshold is assumed to be 0 (the transition could be fired in any case) globally; the probability of every state is assumed to be 0.99; and the credibility of the backup system place is assumed to be 0.6 globally.

The two-stage optimization algorithm for the fuzzy weight vector was simulated using MATLAB.

The fitness function of the proposed two-stage optimization algorithm is shown in fig. 4 and fig. 5. As shown by the simulation results, the optimization converge speed of the second stage is significantly faster than that of the first stage and is very stable upon convergence. Thus, the two stages of this algorithm can be used to improve the optimization convergence speed; furthermore, the speed is very stable upon convergence.

Moreover, from the simulation process, the convergence speed is very fast, as shown in fig. 5, in the general case, and the optimized result could be obtained within 90 steps. Therefore, accelerating the convergence speed and avoiding the premature convergence aspects of the two-stage optimization algorithm are advantages of this method. The simulation shows that the convergence speed is high, as the optimization results were obtained in approximately 90 steps, which is shown as fig. 6.

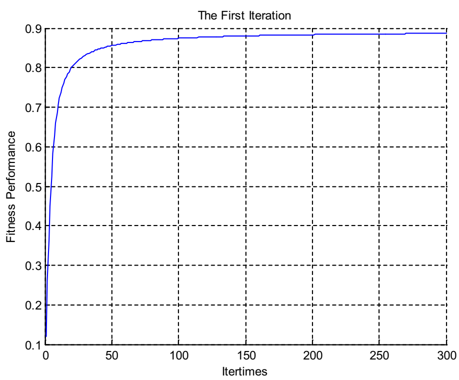

Fig. 4. The first iteration

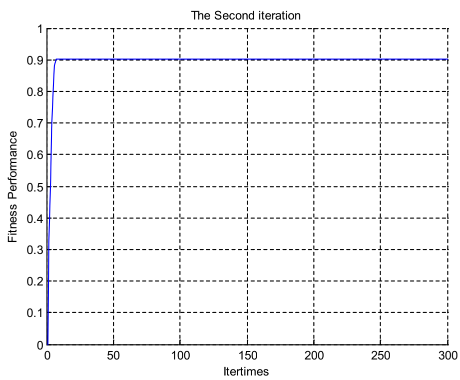

Fig. 5. The second iteration

In addition, for the 300 weights generated by this algorithm, the mean absolute error sum was 0.0077247, and representatives of the fuzzy-weighted values are listed in table 2. In table 2, the performance level is expressed as the contribution ratio, which is the contribution of different states of the product to the whole system when the system is in its normal state.

According to formula (8), the fuzzy general generating function of each component in the system can be expressed in the following form:

Component 1:

$$u_1(z) = \tilde{p}_{11} z^{\tilde{g}_{11}} + \tilde{p}_{12} z^{\tilde{g}_{12}} + \tilde{p}_{13} z^{\tilde{g}_{13}}$$

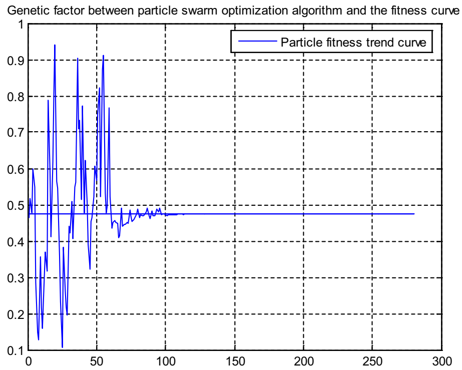

Fig. 6. Number of iterations and particle fitness

Table 2. Representative value of the fuzzy weighted values and performance rates

j	**1**	2	3
$\tilde{\omega}_{1j}$	(0.195, 0.19, 0.185)	(0.295, 0.29, 0.287)	(0.032, 0.03, 0.025)
$\tilde{\omega}_{ij}$	(0.895, 0.89, 0.885)	(0.795, 0.79, 0.785)	(0.955, 0.95, 0.949)
$\tilde{\omega}_{1'j}$	(0.894, 0.89, 0.888)	(0.9, 0.8, 0.88)	/
$\tilde{g}_{1j}$	1.5	2	4
$\tilde{g}_{ij}$	(0.9, 1, 1.1)	(1.4, 1.5, 1.7)	0
$\tilde{g}_{1'j}$	0	0	/

$$
\begin{aligned}
&= (\alpha(d_{11}, t_1) - \tilde{\omega}_{11}) z^{\tilde{g}_{11}} + (\alpha(d_{12}, t_2) - \tilde{\omega}_{12}) z^{\tilde{g}_{12}} + (\alpha(d_{13}, t_3) - \tilde{\omega}_{13}) z^{\tilde{g}_{13}} \\
&= ((0.99 - 0.195), (0.99 - 0.19), (0.99 - 0.185)) z^{1.5} \\
&\quad + ((0.99 - 0.895), (0.99 - 0.89), (0.99 - 0.885)) z^{(0.9,1,1.1)} \\
&\quad + ((0.99 - 0.894), (0.99 - 0.89), (0.99 - 0.888)) z^{0} \\
&= (0.795, 0.8, 0.805) z^{1.5} + (0.095, 0.1, 0.105) z^{1} + (0.096, 0.1, 0.102) z^{0} \qquad (18)
\end{aligned}
$$

Similarly, for Component i:

$$
\begin{aligned}
u_i(z) = {} & (0.695, 0.7, 0.703) z^{2} + (0.195, 0.2, 0.205) z^{(1.4,1.5,1.7)} \\
& + (0.09, 0.1, 0.11) z^{0} \qquad (19)
\end{aligned}
$$

Component 1':

$$
u_{1'}(z) = (0.958, 0.96, 0.965) z^{4} + (0.035, 0.04, 0.041) z^{0}
$$

Based on the WFPNs addition algorithm, as shown in fig. 2, component 1 and component i are constructed in parallel, and component 1' is constructed in series. The generating function of the system is as follows:

$$
U(z) = \tilde{\varphi}_s(\tilde{\varphi}_p(u_1(z), u_i(z)), u_{1'}(z))
$$

$$\begin{aligned}
&= (0.52932, 0.5376, 0.54611)z^{3.5} \\
&\quad + (0.14851, 0.1536, 0.15925)z^{(2.9,3,3.2)} \\
&\quad + (0.063252, 0.0672, 0.071231)z^{(2.9,3,3.1)} \\
&\quad + (0.017747, 0.0192, 0.020772)z^{(2.3,2.5,2.8)} \\
&\quad + (0.063918, 0.0672, 0.069196)z^{2} \\
&\quad + (0.017934, 0.0192, 0.020178)z^{(1.4,1.5,1.7)} \\
&\quad + (0.068545, 0.0768, 0.085451)z^{1.5} \\
&\quad + (0.008191, 0.0096, 0.01115)z^{(0.9,1,1.1)} \\
&\quad + (0.042097, 0.0496, 0.053066)z^{0}
\end{aligned} \tag{20}$$

Assume all the states are successful. For the successful states $\tilde{\omega}_i < \tilde{g}_i$, the fuzzy availability is calculated as

$$\tilde{A}(\tilde{\omega}) = \tilde{\delta}_A(U(z), \tilde{\omega}) = (0.90743, 0.93888, 0.97017) \tag{21}$$

Suppose that the system safety standard requires that the system operation must satisfy a required level of system availability, which is set as 0.9. After evaluation, the above system design, considering fuzzy uncertainties, can satisfactorily meet the system availability requirements, which guarantees that the system is working in a relatively safe manner.

We compared the improved algorithm of the WFPNs with the learning ability of the FPNs of the non- improved algorithm, and the comparison verified the accuracy and training speed of the algorithm. To contrast the improved algorithm of the WFPNs with the learning ability of the FPNs of the non- improved algorithm, the simulation value was set to 400 sample data points. The average error is shown in Table 3.

Table 3. Average error

Algorithm	Average error
The improved algorithm of the WFPNs	0.8e-05
The non-improved algorithm of the WFPNs	4.2e-05

Table 3 illustrates that the average error is smaller and that the prediction accuracy is higher than the non- improved algorithm. These values can be used to determine the convergence precision and stability. This comparison demonstrates that compared to the non- improved algorithm, the learning algorithm of the model depends more heavily on the position and velocity of the learning parameters, has more specific importance and dependence information, and has a faster convergence speed. In addition, this paper proposes an adaptive learning algorithm that disturbs every particle using the adaptive function and improves the diversity of the particle swarm to a certain extent. The magnitude of the perturbation decreases as additional iterations are performed, which guarantees convergence.

6 SUMMARY

In this paper, the WFPNs model of a fuzzy multi-state system is established, whereby the weight is able to learn based on an improved particle algorithm. The WFPNs has the following characteristics: First, the place denotes the component of the fuzzy multi-state system, and every place has a finite set of tokens, which represent the fuzzy state. Second, each token has a different probability for different transitions and the likelihood of transitions. Third, for every transition, the degree of support from one place to different transitions is different. The probability of every initial state is assumed as a constant; subsequently, the weights of the fuzzy state are estimated using an improved particle algorithm, and the probability is changed as the weights. In the improved particle approach, a two-stage optimization strategy is employed to ensure that the weights are more accurate. Finally, the transient-state reliability is evaluated according to the weights, which are based on the FUGF. MATLAB simulations demonstrate that the proposed reliability evaluation method based on the weight vectors of the WFPNs is more accurate than previous methods and thus provides theoretical guidance for system reliability evaluation, which can also be used to assess a possible fuzzy multi-state design under uncertainties over if it can meet the desired reliability requirements. Reliability performance will have great significance in the further study of fuzzy multi-state robustness and fuzzy uncertainty, which will be the focus of future research.

Future work:

Another possible solution is to apply the adaptive capability of genetic algorithm in helping adjust the threshold values and certainty factors. By mapping fuzzy production rules into a multilayer perceptron neural network and using this neural network's adaptive and learning capability, threshold values and certainty value can be adjusted more efficiently and accurately.

REFERENCES

[1] Qi Wang. The utility fuzzy mathematics. Beijing: Science and Technology Literature Press, 1992.

[2] Chang, K. P., D. Chang, and E. Zio. "Application of Monte Carlo Simulation for the Estimation of Production Availability in Offshore Installations". In Simulation Methods for Reliability and Availability of Complex Systems, Chapter 8, pp.233 – 252. Springer, 2010..

[3] Volovoi, V., and D. K. Peterson. 2011, December. "Coupling Reliability and Logistical Considerations for Complex System of Systems using Stochastic Petri Nets". In Proceedings of the 2011 Winter Simulation Conferencepp.1751-1762. Piscataway, New Jersey2011.

[4] Murata, T. Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE. Vol.77, no. 4, pp.541-579, April 1989.

[5] Murata, V, S. Subrahmanian and T.Wakayama. A Petri nets model for Reasoning in the Presence of Inconsistency. IEEE Transactions on Knowledge and Data Engineering. Vol.3, no.3, pp.281-292, September 1991.

[6] Bechtold, R. Knowledge Attributes:Fuzzy Application of Temporal Constraints in Active Expert Database Systems. Proceedings of the IEEE/ACM forth International Conference on Developing and Managing Expert Systems. Los Alaminos, CAp. p.29-35, October 1991.

[7] Valette R.Cardoso J.Dubois D. Monitoring manufacturing systems by Petri nets with imprecise marking. IEEE International Symposium on Intelligent Control. no.9, pp. 233-238, September, 1989.

[8] Looney C G. Fuzzy Petri nets and applications. In: Tzafestas S. G. et al. Fuzzy Reasoning in Information. Decision and Control Systems. 1994, 511-527.

[9] Jumei Yuan, ChaozhenHou. Stochastic Petri nets fuzzy evaluation of repairable system availability. Computer Engineering, Vol.33, no.8, pp.17-19, April 2007.

[10] Peiming Bao.Based on BP network learning ability of fuzzy Petri nets. Computer Engineering. Vol.27, no.5, pp.695-702, February 2004.

[11] Li Jiao and To-Yat Cheung. Characterizing Liveness Monotonicity for Weighted Petri Nets in Terms of Siphon-Based Properties. International Journal of Foundations of Computer Science. Vol.14, no.4, pp.641-658, August 2003, 14(4): 641-658.

[12] HAMMADI BENNOUI, ALLAOUA CHAOUI, KAMEL BARKAOUI. ON STRUCTURAL ANALYSIS OF INTERACTING BEHAVIORAL PETRI NETS FOR DISTRIBUTED CAUSAL MODEL-BASED DIAGNOSIS. International Journal of Foundations of Computer Science. Vol.23, no.7, pp.1523-1541, November 2012.

[13] Y.Ding, A.Listrianski. Fuzzy universal generating functions for multi-state system reliability assessment. Fuzzy Sets and Systems, Vol.159, no.3, pp. 307-324, June 2008.

[14] G.Levitin.The universal generating function in reliability analysis and optimization.London: Springer, 2005.

[15] A.Lisnianski, G.Levltin.Multi-state system reliability assessment, optimization, applicationS. ingapore: World Scientific, 2003.

[16] P.A.V. de Miranda, A.X. Falc, J.K. Udupa. Synergistic arc-weight estimation for interactive image segmentation using graphs. Vol.114, no.1, pp.85-99, January 2010.

[17] Z.G.Tian, G.Levitin, M.J.Zuo.A joint reliability-redundancy optimization approach for multi-state series-parallel systems. Reliability Engineering and System Safety, Vol.94, no.10, pp.1568-1576, October 2009.

[18] G.Levitin. Intelligence in reliability engineering, new metaheuristics, neural and fuzzy techniques in reliability. London:Springer, 2007.

[19]]R.Meziane, Y.Massim, A.Zeblah, et al. Reliability optimization using ant colony algorithm under performance and cost constraints. Electric Power Systems Research, Vol.76, no.1-3, pp.1-8, Septembe 2005.

[20] V.K.Sharma, M.Agarwal, K.Sen.Optimal structure in heterogeneous multi-state series-parallel reliability systems. Economic Quality Control, Vol.25, no.1, pp.127-150, August 2010.

[21] Shen VRL. Reinforcement learning for high-level fuzzy Petri netsI. EEE Transactions on System, Man, and Cybernetics-Part B 2003; Vol.33, no.1, pp.351-361, Apirl 2003.

[22] K.Y. Cai, Introduction to Fuzzy Reliability, Kluwer Academic Publishers, Dordrecht, 1996.

[23] A. Kaufmann, M.M. Gupta, Fuzzy Mathematical Models in Engineering and Management Science, North-Holland, Amsterdam, 1988.

[24] H.Motameni, T.Ghassempouri. Transforming Fuzzy Communication Diagram to Fuzzy Petri net[J]. American Journal of Scientific, ISSN1450-223X, EuroJournals Publishing, 2011.

[25] Chen S M. Weighted fuzzy reasoning using weighted fuzzy Petri nets[J]. IEEE Trans.on Knowledge and Data Engineering, Vol.14, no.2, pp.386-397, March 2002.

[26] Tian Li-qing, Lin-Chuang, Zhou Wen-jiang. Performance Eguivalent Simplification of Seguent and Parallel Transitions in Stochastic Petri Nets.ACTA ELECTRONICA SINICA. Vol.30, no.8, pp.1134-1137, January 2002.

[27] Holger Hermannst Ulrich Herzog Vassilis Mertsiotakis etc. Exploiting Stochastic Process Algebra Achievementsfor Generalized Stochastic Petri Nets. Saint Malo. no.3-6, pp.183-192, June, 1997.

[28] Levifin G. Reliability evaluation for acyclic transrnission networks of multi-state elements with delays. IEEE Transaction on Reliability, 2003, 81(1):1-7.

[29] Shaikh, A., and A. Mettas. 2010. "Application of Reliability, Availability, and Maintainability Simulation to Process Industries: a Case Study". In Simulation Methods for Reliability and Availability of Complex Systems, Chapter 8, pp.173 – 197. Springer., 2010

Zhang Xinju Ph.D., Beihang University. Her main research directions are formal modeling and verification. BeiHang University 100191 Beijing, China juzi200501@163.com

Yao Shuzhen Professor, doctoral tutor, member of the Software Engineering Teaching Guidance Committee in Ministry of Education, executive director of the China Software Industry Association, group leader of the National Coordination Group of Master of Engineering in the field of software engineering. Her research interests include distributed computing, formal modeling and verification, software development environments, and network information security.