J Med Syst (2012) 36:1283–1300
DOI 10.1007/s10916-010-9589-6

ORIGINAL PAPER

Variances Handling Method of Clinical Pathways Based on T-S Fuzzy Neural Networks with Novel Hybrid Learning Algorithm

Gang Du · Zhibin Jiang · Xiaodi Diao · Yan Ye · Yang Yao

Received: 26 April 2010 / Accepted: 1 September 2010 / Published online: 23 September 2010

Abstract Clinical pathways' variances present complex, fuzzy, uncertain and high-risk characteristics. They could cause complicating diseases or even endanger patients' life if not handled effectively. In order to improve the accuracy and efficiency of variances handling by Takagi-Sugeno (T-S) fuzzy neural networks (FNNs), a new variances handling method for clinical pathways (CPs) is proposed in this study, which is based on T-S FNNs with novel hybrid learning algorithm. And the optimal structure and parameters can be achieved simultaneously by integrating the random cooperative decomposing particle swarm optimization algorithm (RCDPSO) and discrete binary version of PSO (DPSO) algorithm. Finally, a case study on liver poisoning of osteosarcoma preoperative chemotherapy CP is used to validate the proposed method. The result demonstrates that T-S FNNs based on the proposed algorithm achieves superior performances in efficiency, precision, and generalization ability to standard T-S FNNs, Mamdani FNNs and T-S FNNs based on other algorithms (CPSO and PSO) for variances handling of CPs.

G. Du · Z. Jiang (✉) · Y. Ye
Department of Industrial Engineering & Logistics Management,
School of Mechanical Engineering,
Shanghai Jiao Tong University,
800 Dong Chuan Road,
Shanghai 200240, China
e-mail: zbjiang@sjtu.edu.cn

X. Diao
Shanghai Putuo District Central Hospital,
Shanghai 200062, China

Y. Yao
Shanghai No. 6 People's Hospital,
Shanghai 200233, China

Keywords T-S fuzzy neural networks · Clinical pathway · Particle swarm optimization (PSO) · Variances handling

Introduction

Clinical pathways (CPs) have been accepted as an important tool to facilitate the delivery of high quality, cost-efficient health care service along with increasingly diverse health demands. The CP is a management plan that displays goals for patients and provides the sequence and timing of actions necessary to achieve these goals with optimal efficiency [1, 2]. Most of references demonstrate that the CP has many advantages for the application in hospitals, such as reducing medical expense, increasing patient satisfaction and improving medical care quality, etc [3, 4].

Although the CP predefines predictable standardized care process for a particular diagnosis or procedure, many variances may still unavoidably occur due to individual complexities and subjective initiatives of both patients/families and healthcare professionals after applying it to patients care. During the process of treatment, the variance handling is the most critical issue in managing and controlling CPs. In general, the CP variances include patient, physician, system, pre-admission, and discharge variances. For example, during osteosarcoma preoperative chemotherapy CP, some uncertain variances (such as white and red blood cells decreasing, platelets decreasing, liver, kidney damage, stomatitis, and neuropathy etc.,) may usually occur [5]. Moreover, most variances present complex, fuzzy, uncertain, and high-risk characteristics, and thus may cause complicating diseases or even endanger patients' life. Therefore, it is necessary to handle these variances accurately and effectively. And the uncertain variances of CPs may need different variance handling measures. In this paper, we pay more attention to one kind of variance, and the type of variance handling measures can be determined in some extent. However, the degree of different variances handling measures is fuzzy. Therefore, we adopt “fuzzy” method instead of “stochastic” method. In real world, some key index parameters have strong relationship with variances handling measures of CPs, and variances handling problems are highly nonlinear in nature so that it’s hard to develop a comprehensive mathematic model.

According to Table 1, previous researches mainly focus on the definition of CP variances (shown in Table 1), and recent researches mainly focus on documenting, classifying, and analyzing occurred variances during the implementation of CPs, e.g., in [12].

As usual, a difficulty in dealing with variances is in its fuzzy and uncertain features. That is to say, during the CP variances handling, many data are imprecise and incomplete. The fuzzy models can describe vague statements as in natural language. Therefore, to cope with this difficulty, a typed fuzzy Petri net extended by process knowledge (TFPN-PK) model method is presented to provide support for the handling decision of different types of CP variances (shown in Table 1), and the different types of variances can be dealt with based on different strategies [13–15]. Moreover, an extended event-condition-action (ECA) rules method is proposed to model and handle the CP variances validly [16].

However, the above methods mentioned in Table 1 depend on the fuzzy rules provided by experts, which are usually difficult to obtain. Moreover, fuzzy systems do not have much learning capability, and it is difficult for a human operator to tune the fuzzy rules and membership functions from the training data.

Artificial neural networks (ANNs) have emerged as important tools for non-linear data mapping [17]. ANNs are useful in solving non-linear problems where the algorithm or rules to solve the problem are unknown or difficult to express. The data structure and non-linear computations of ANNs allow good fits to complex, multivariable data. ANNs process information in parallel and are robust to data errors. A disadvantage of ANNs is regarded as a “black-box” technology because of the lack of comprehensibility. Therefore, although ANNs are capable of learning the complex nonlinear relationships between index parameters and CP variances handling, they cannot explicitly provide the readable and comprehensible rules for healthcare professionals to make clinical decisions. Moreover, it is much difficult by using ANNs to handle the fuzzy and uncertain CP variances.

Fuzzy neural networks (FNNs) combine the advantages of both fuzzy logic in processing vague and uncertain information and neural network in good learning abilities,

Table 1 Summary of main studies reviewed on CP variances

Author(s) and year	Main country studied	Clinical area	Brief description/key findings
Wigfield and Boon (1996) [6]	–	–	“Variances are differences between what is expected to happen and what does happen”
Cheah (1998) [7]	Singapore	Personal perspective	Examines the growth and benefits implications of pathways
M.B. Price, A. Jones, J.A. Hawkins, et al.,(1999) [8]	–	–	Variances are considered as any deviations from a CP that occur during the patient care process based on the standardized pathway
Atwal and Caldwell (2002) [9]	–	–	The definition of variance “any deviation from the proposed standard of care listed in the pathway”.
Bryan et al. (2002) [10]	England, UK	Breast cancer	Describes the key steps in developing a variance, and considers issues of task distribution and de-skilling
Caminiti et al. (2005) [11]	Italy	Various	Evaluates the effectiveness of a multifaceted strategy to reduce negative variations in medical practice
Wakamiya S., and Yamauchi, K., (2006) [12]	Japan	Various	documenting, classifying, and analyzing occurred variances during the implementation of CPs
Ye Y., Jiang Z.B., Yang D., and Du G., (2008) [13]	China	Various	A TFPN-PK model method is presented to provide support for the handling decision of different types of clinical pathway variances
Ye Y., Jiang Z.B., Diao X.D., and Du G., (2009) [14]	China	Various	Adopting a fuzzy rule-based reasoning method to handle variances of CPs
Ye Y., Jiang Z.B., Diao X.D., and Du G., (2009) [15]	China	Various	The different types of variances can be dealt with based on different strategies
Du G., Jiang Z.B., Diao X.D., Sun Y.J, Ye Y., and Yao Y., (2009) [16]	China	Various	A extended event-condition-action (ECA) rules is proposed to model and handle the osteosarcoma preoperative chemotherapy CP variances

meanwhile, overcome their weaknesses [18]. Fuzzy neural network can also handle imprecise information through linguistic expressions. For several decades, FNNs have attracted much attention and have been applied in many domains [19]. Two typical types of FNNs are Mamdani-type and Takagi-Sugeno (T-S)-type models [20, 21]. Many researches have shown that if a T-S fuzzy model is used, the network size and learning accuracy is superior to those of Mamdani-type fuzzy model [22].

The accuracy and efficiency of variances handling for CPs are very important. According to the characteristics of CPs' variances, T-S FNNs can be used for variances handling of CPs in assisting doctors to make decisions. However, there are many drawbacks of T-S FNNs by commonly back propagation (BP) algorithm with gradient descent for training [23], such as being easy to trap into local minimum point, being sensitive to initial values and poor ability on global search. Therefore, many researches use genetic algorithms (GAs) for the learning of the fuzzy models, and attain better performance than BP algorithm [24]. For example, hybrid of GA and Kalman filter [25], and multi-objective hierarchical GA [26] have been proposed to optimize fuzzy models.

In the [27], GA algorithm is used to adjust the parameters of T-S fuzzy neural network. Lin and Xu [28] built T-S neural network with consequent parts as constants. The parameters of membership functions and consequent part of the network were coded and an improved GA was used to search for the optimum parameters. However, GA is characterized by huge computation time and slow convergence near the optimum [29].

Recently, a new evolutionary technique, the particle swarm optimization (PSO) [30] has emerged as an important optimization technique [31, 32]. Compared with GA, the PSO has some attractive characteristics: (1) it has a fast convergence rate; (2) it is easy to implement; (3) it has few parameters to adjust. It retains good knowledge studied by all particles. And it encourages cooperative co-evolution and information sharing between particles, which enhance the search for a global optimal solution. Successful applications of the PSO to some optimization problems, such as function optimization and neural network optimization, have demonstrated its potential [30, 32]. The framework for identifying the fuzzy models using PSO algorithm is proposed by [33, 34]. In addition, the authors found that with the same complexity that was generated from the same data, PSO algorithm can generate better results for identifying the fuzzy models than GA algorithm for fuzzy neural network optimization.

For example, Shoorehdeli and Teshnehlab [35] reported a hybrid learning algorithm which is based on particle swarm optimization for training the antecedent part and gradient descent for training the conclusion part.

Although many researches have shown that the PSO performs well for global search because of its capability in quickly finding and exploring promising regions in the search space, they are relatively inefficient in local search and easy to result in premature convergence. Therefore, some improved approaches and variants of PSO have been reported in [36–40]. These methods improve the global ability of the PSO to some degree, but they are hard to get a good tradeoff between global convergence and convergent efficiency.

Cheng-Jian Lin et al. [41, 42] proposed the improving particle swarm optimization and immune-based symbiotic particle swarm optimization for use in T-S fuzzy neural networks can improve the ability of searching global solution, thus the accuracy can be improved. However, it cost long computation time and there is a weak ability in high dimension optimization problem.

Combined with the notion of co-evolution, a cooperative particle swarm optimization (CPSO) algorithm [43] has been proposed and proven to be more effective than the traditional PSO in most optimization problems. For example, Ben Niua, et al. [44] adopted the multi-swarm cooperative PSO to optimize the parameters of the fuzzy neural networks, and obtained higher accuracy with a large computation cost. However, for this method, there is a weak ability in the structure and parameters optimization simultaneously. The weak and strength sides of all methods mentioned are summarized in Table 2.

As shown in Table 2, computational intelligence significantly enhances the accuracy. However, most of the above mentioned methods have their shortcomings: (1) few optimize simultaneously the structure and parameters of T-S FNNs; (2) inefficiency in optimize the high dimensional nonlinear parameters; (3) time-consuming. Furthermore, most of the methods are easy to result in premature convergence.

As usual, the CPSO algorithm with a simple splitting and collaborating approach may become trapped in the sub-optimal locations in the search space. The sequence of the subgroup optimization can influence its optimization result. Moreover, the original PSO uses a population of n-dimensional vectors, and then these vectors can be partitioned into n swarms of 1-D vectors for the CPSO algorithm. However, with the increasing of the dimensions of the population, the computing time will increase greatly. If an n-dimensional vector is split into K parts roughly (K called the split factor) in order to decrease the computing time, every parameter in the n-dimensional vector will not be optimized sufficiently. In addition, many researchers adopt random perturbation and existent mutation mechanisms, e.g., in [45] to avoid comparative inefficiency of the CPSO in fine tuning the solution. However, this problem cannot be resolved radically.

In this study, a subgroup decomposing method is adopted for the particles in random cooperative evolution

Table 2 The comparison table on weak and strength sides of mentioned methods

Method	Learning algorithm		Advantages	Disadvantages
	The antecedent part	the consequent part		
Fuzzy neural [20] networks (Mamdani)	back-propagation (BP) algorithm	fuzzy reasoning	Common	The defuzzification process is complex
(Takagi–Sugeno)[21] FNNs	BP algorithm	linear functions of inputs	fewer lines	Low accuracy and instability
ANFIS[22]	BP algorithm	a linear least-squares estimation	Hybrid learning procedure	Low accuracy
Oh, et al. [24]	The same to standard FNNs	Genetic algorithm (GA)	Improve accuracy	Large computation cost
Wang, et al. [25]	Genetic algorithm (GA)	Kalman filter	Improve accuracy	Slow convergence near the optimum
Wang, et al. [26]	Multi-objective hierarchical GA	Multi-objective hierarchical GA	Improve accuracy	Huge computation time
Tang, Quek, and Ng(2005) [27]	Genetic algorithm (GA)	linear function of inputs	Improve accuracy	Space distribution of samples is not considered
Lin and Xu (2006) [28]	Genetic algorithm (GA)	constants	Improve accuracy	Large computation cost
Khosla, A., Kumar, S., and Aggarwal, K.K., [33, 34]	Particle swarm optimization	Particle swarm optimization	Decrease computation time	Lacking in integrated optimization
Shoorehdeli and Teshnehlab [35]	particle swarm optimization	gradient descent	Improving performance	Complex and hard to use
Cheng-Jian Lin et al (2009) [41]	Improving Particle swarm optimization (PSO)	Improving Particle swarm optimization	Improve accuracy	Weak ability in high dimension optimization
Cheng-Jian Lin (2008) [42]	Immune-based symbiotic PSO	immune-based symbiotic PSO	Improve global convergence	Long computation time.
Begrgh and Engelbrecht [43]	Cooperative particle swarm optimization	a linear least-squares estimation	Improve accuracy	Large computation cost
Ben Niua, et al. (2008) [44]	multi-swarm cooperative PSO	multi-swarm cooperative PSO	much higher convergence speed and stability	Weak in the structure and parameters optimization simultaneously

with the highest performance, and the "exponential smoothing mutation mechanism" is used for avoiding comparative inefficiency and local searching to further improve the quality of solutions. This method is much different from the existing ones, and hopefully it can increase the diversity of the swarms, and avoid plunging into local optimum. This method is called as the random cooperative decomposing particle swarm optimization algorithm (RCDPSO). Moreover, a discrete binary version of PSO (DPSO) by Kennedy and Eberhart [46] is employed to optimize the T-S FNNs structure. Then, a new variances handling method of CPs is proposed, which is based on T-S FNNs with integrating RCDPSO/DPSO optimization to improve the efficiency, precision, and generalization ability of T-S FNNs for variances handling of CPs. Especially, this study puts its emphasis on using RCDPSO/DPSO to optimize simultaneously the structure and parameters of T-S FNNs, and it is much different from those previously researches.

The remainder of this paper is organized as follows. "The review of particle swarm algorithm" makes review on particle swarm algorithm. "Random cooperative decomposing particle swarm optimization algorithm" describes the principle of RCDPSO algorithm. "A novel hybrid learning algorithm of T-S fuzzy model for variances handling of CPs" gives a novel hybrid learning algorithm of T-S FNNs for variances handling of CPs. A case study on liver poisoning of osteosarcoma preoperative chemotherapy is used to validate the proposed method in "Case study". The whole article is summarized and the further research is sketched in "Conclusion".

The review of particle swarm algorithm

The PSO algorithm is an adaptive algorithm based on a social-psychological metaphor. Each particle has its own position and velocity to move around the search space. The *j*th component of the *i*th particle represents a possible solution and has a position vector x_{ij}, a velocity vector v_{ij}, and the best personal position p_{ij} encountered so far by the particle. During each iteration, each particle accelerates in the direction of its own personal best position p_{ij}, as well as in the direction of the global best position p_{gj} discovered so far by any of the particles in the population. At each time step *t*, the velocity of the *j*th component of the *i*th particle is updated using Eq. 1:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_1\left(p_{ij}(t) - x_{ij}(t)\right) + c_2r_2\left(p_{gj}(t) - x_{ij}(t)\right) \tag{1}$$

where w is the inertia weight; c_1 and c_2 are acceleration coefficients; r_1 and r_2 are uniformly distributed random numbers in the range (0, 1). The velocity v_{ij} is limited to the range [v_{min}, v_{max}]. Based on the updated velocities, the *j*th component of the *i*th particle is calculated using

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{2}$$

Based on (1) and (2), the populations of particles tend to cluster together with each particle moving randomly in a random direction.

Kennedy and Eberhart proposed a discrete binary version of PSO [46], which is very appropriate to solve discrete optimization problems. DPSO takes the values of binary vectors of length n and the velocity represents the probability that a bit will take the value 1. The velocity updating formula Eq. 1 remains unchanged, but it is constrained to the interval [0, 1] by a limiting transformation S(v_{ij}). Then the particle changes its bit value by the following Equations:

$$S\left(v_{ij}\right) = 1/\left(1 + \exp\left(-v_{ij}\right)\right) \tag{3}$$

$$P_{ij} = \begin{cases} 1 \; rand \; < S\left(v_{ij}\right) \\ 0 \; otherwise \end{cases} \tag{4}$$

where v_{ij} is the j^{th} velocity dimension of particle i, p_{ij} is the j^{th} position dimension of particle i, S(v_{ij}) is a constrain convert function and *rand* is a random number drawn from U(0,1).

Random cooperative decomposing particle swarm optimization algorithm

The PSO algorithm may easily get trapped in a local optimum when tackling complex problems. Through empirical studies, it has observed that combing the CPSO algorithm and PSO algorithm can get a better result. The detailed description can be shown in [43].

In order to avoid comparative inefficiency of the CPSO in finely tuning the solution and further obtain better solutions gotten by CPSO algorithm, the "exponential smoothing mutation mechanism" is used once it satisfies the predefined iteration cycle. That is to say, implementing local searching by small mutation is to further improve these solutions found by the RCDPSO algorithm. By this method, a better solution can be found. Then, by means of combing the current personal information and the best personal history information of the particles, the new particle parameters will be gotten.

During the process of cooperative evolution with random execution sequence, the decomposing algorithm is adopted for particles with the best performance. The method for evaluating the performance of particles is given as follows:

We can make the evaluation for each m number of iterations. During this period, the deviation value between the fitness value of the particle in each subgroup and the best global particle in iteration is regarded as a judging criterion. The smallest deviation value is regarded as the best subgroup, and then it would be further decomposed. The decision formula is

$$\begin{aligned} \varepsilon_{k,j} = {} & [f\_(1,j,k) - fgbest\_(j,k)] \\ & + [f\_(2,j,k) - fgbest\_(j,k)] + ,\ldots, \\ & + [f\_(r,j,k) - fgbest\_(j,k)] \end{aligned} \tag{5}$$

where, j is the j^{th} subgroup, r means the particle, f(r, j, k) is the r^{th} particle's fitness value of the j^{th} subgroup in the k^{th} iteration with co-evolution. fgbest_(j, k) is the best fitness value of the j^{th} subgroup in the k^{th} iteration. $\varepsilon_{k,j}$ is the deviation sum of the global best particle of the j^{th} subgroup in the k^{th} iteration. During a period, the average of the deviation sum of the subgroup can be given as follows:

$$P_{l,j} = \left(\sum_{k=(l-1)^* life+1}^{l^* life} \varepsilon_{k,j} \right) / life \tag{6}$$

Here *l* means the period, and life is a length of the period.

The "exponential smoothing mutation" is used for local searching to further improve the quality of solutions. The principle of the "exponential smoothing mutation mechanism" is to offer the guidance for optimization by combing the current personal information and the best personal history information of the particles. The "m" current best performance particles and "m" personal best particles recorded in the history warehouse are taken out. And these particles are regarded as the "base" of the exponential smoothing. Then the new parameter structure can be defined, which is named as the "weight". The weight of the dimensionality is the same as the parameters for optimization, so the dimension of "posa" is the same as the one of "pos". It can be defined as follows:

$$posa = rand(popa\_\ size, parts\_\ size) \tag{7}$$

The "posa" is also given a rand number with (0, 1), here, popa_size denotes the particles, and parts_size denotes the dimensionality of the parameters. The posa(i, j) means that the i^{th} particle with the j^{th} parameter.

$$i \in [1, popa\_\ size]; j \in [1, parts\_\ size]$$

The pseudocode of the "exponential smoothing mutation mechanism" can be given as follows:

```
For each particle i∈[1,...,popa_size]
   For each dimensional parameter j∈[1,...,parts_size]
      F(i, j)= $a_1$(i, j)×pos(1, j)+,…,+ $a_m$(i, j)×pos(m, j)
      //The pos(i, j) is m current best pest performance particles. The F(i, j)
        is the expect value of the $j^{th}$ wight of the $i^{th}$ particle based on "pso".
        pF(i, j)= $a_1$(i, j)×pbest(1, j)+,…,+ $a_m$(i, j)×pbest(m, j)
     // The pbest(i, j) is m personal best particles in the history warehouse.
        The pF(i, j) is the expected value calculated from the $j^{th}$ weight of the
        $i^{th}$ particle based on "pbest".
      End
End
```

Here, a_k(i, j) can be obtained by the following exponential smoothing equation:

$$a_k(i,j) = \alpha(i,j) \times [1 - \alpha(i,j)]^k, k = 1, 2, \ldots, m \qquad (8)$$

The pseudocode of the normalized a_k(i, j) can be given as follows:

```
For i=1:popa_size (each particle)
    For j=1:parts_size (each dimensional parameter)
         For k=1:m
              tempapha(k, j, i)=posa(i, j)*(1-posa(i, j)).^k;
         End
              sumapha(i, j)=sum(tempapha(:, j, i));
         For k=1:m
              weight(k, j, i)=tempapha(k, j, i)/sumapha(i, j);
         End
    End
End
```

The normalized a_k(i, j) is called weight, m is regarded as the "base" of the corresponding particle number in the "pos", and the initial value of "a(i, j)" is a random matrix of m×n with (0, 1].

$$a(i,j) = rand(popa\_\ size, parts\_\ size)$$

After the parameters of the "posa" having been optimized for a few times, we take out n best particles from the "posa", and calculate the corresponding shape parameter of the trapezoid membership function. Then the n particles are used as "the mutation particles" to rewrite the previous n particles with the worse performance obtained by RCDPSO optimization.

A novel hybrid learning algorithm of T-S fuzzy model for variances handling of CPs

The Structure of T-S FNNs

For the multiple-input and single-output (MISO) system, the T-S model [20] can be given as follows: Let X = [x_1, x_2, ..., x_n]^T denote an input vector, where each variable x_i is a fuzzy linguistic variable. The set of linguistic variables for x_i is represented by $T(x_i) = \{A_i^1, A_i^2, \ldots, A_i^{m_i}\}$, $i = 1, 2, \ldots, n$, where $A_i^{r_i}(r_i = 1, 2, \ldots, m_i)$ is the r_i^{th} linguistic value of the input x_i. The membership function of fuzzy set defined on domain of x_i is

$\mu_{A_i^{r_i}}(x_i)(r_i = 1, 2, \ldots, m_i)$. Let y denote output vector. A rule takes form of

$$\begin{aligned} &Rule_j : If\, x_1 \text{ is } A_1^{jr_1} \text{ and } x_2 \text{ is } A_2^{jr_2} \text{ and} \cdots \text{ and } x_n \text{ is } A_n^{jr_n}, \\ &\quad Then\ \mathrm{y}^j = p_{_0}^j + p_{_1}^j x_1 + \cdots + p_{_n}^j x_n \\ &\qquad j = 1, 2, \ldots, m, m \le \prod_{i=1}^{n} m_i \end{aligned} \tag{9}$$

According to [20, 33, 35], the T-S FNNs is composed of five layers, which are divided into two networks: antecedent and consequent. The first four layer of this T-S FNNs correspond to the antecedent network, the fifth layer is output layer, and the consequent parts are indicated with broken line.

Layers 1 (Input layer): No function is performed in this layer. The node only transmits input values to layer 2.

Layers 2 (Fuzzification layer): Nodes in this layer correspond to one linguistic label of the input variables in layer 1. That is to say, the membership value specifying the degree to which an input value belongs to a fuzzy set is calculated in this layer. In this study, the trapezoid membership function is adopted.

Layers 3 (Rule layer): Firing strength of every rule is calculated. There are m rules. According to the premise parts of rules, the neuron representing some linguistic values of some input is connected with the neuron representing some rule. The degree the input vector x matches rule Rule_j is computed by the min operator: $\alpha_j = \mu_{A_1^{jr_1}}(x_1)\mu_{A_2^{jr_2}}(x_2)\ldots\mu_{A_n^{jr_n}}(x_n)$. α_j is called the firing strength of rule Rule_j.

Layers 4 (Normalized layer): In the fourth layer, the output of each node in this layer is the normalized firing strength. These parameters denote linear parameters or so called consequent parameters. Thus, the outputs of this layer can be given by $\overline{\alpha}_j = \alpha_j / \sum_{j=1}^{m} \alpha_j$.

Layers 5 (Output layer): This layer sums up all the activated values from the inference rules to generate the overall output y.

$$\begin{aligned} y &= \frac{\sum_{j=1}^{m}\left[\left(\prod_{i=1}^{n}\mu_{A_i^{jr_i}}(x_i)\right)\times y^j\right]}{\sum_{j=1}^{m}\prod_{i=1}^{n}{}_{A_i^{jr_i}}(x_i)} = \frac{\sum_{j=1}^{m}\alpha_j y^j}{\sum_{j=1}^{m}\alpha_j} \\ &= \sum_{j=1}^{m}\overline{\alpha}_j y^j, j = 1, 2, \ldots, m \end{aligned} \tag{10}$$

The principle of novel hybrid learning algorithm

T-S FNNs are optimized by adapting the structure, the antecedent parameters (membership function parameters) and consequent parameters so that a specified objective function is minimized. In this study, a novel hybrid algorithm by integrating the RCDPSO/DPSO algorithm and kalman filtering algorithm is proposed to tune the structure and parameters of T-S FNNs for variances handling of CPs. This hybrid procedure is iterated until the output error is reduced to a desired goal or a maximal number of generations predefined by user reached.

Fitness function: The fitness function for measuring the performance of individuals is defined as:

$$\text{MSE} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(T_{ij} - y_{ij}\right)^2, \quad \text{MAE} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left|T_{ij} - y_{ij}\right| \tag{11}$$

where T_{ij} is the expected output, y_{ij} is the predicted output, n is the number of training samples, and m is the number of output nodes. Thus, the fitness is the mean squared error (MSE) of T-S FNNs model. The mean square error (MSE) of training samples and testing samples are calculated respectively to evaluate the performance of the network. The mean absolute error (MAE) is other evaluation performance index for T-S FNNs.

The least square algorithm is usually adopted for the consequent parameters estimation. However it is not feasible when the matrix is a singular matrix or approximate singular matrix. Therefore, kalman filtering algorithm is employed for the consequent parameters estimation [47]. Suppose "m" is the number of rules, and the input set is $X_i = (x_{1i}, x_{2i}, \ldots, x_{ni})^T$, according to the Eq. 10, the output can be given as follows:

$$\begin{aligned} &\overline{\alpha}_i^j = \alpha_i^j / \sum_{j=1}^{m}\alpha_i^j, \\ &y_i = \sum_{j=1}^{m}\overline{\alpha}_i^j y_i^j = \sum_{j=1}^{m}\overline{\alpha}_i^j\left(p_0^j + p_1^j x_{1i} +, \ldots, p_n^j x_{ni}\right) \\ &\quad = \left[\overline{\alpha}_i^1, \overline{\alpha}_i^2, \ldots, \overline{\alpha}_i^m, \overline{\alpha}_i^1 x_{1i}, \overline{\alpha}_i^2 x_{1i}, \ldots, \overline{\alpha}_i^m x_{1i}, \overline{\alpha}_i^1 x_{ni}, \overline{\alpha}_i^2 x_{ni}, \ldots, \overline{\alpha}_i^m x_{ni}\right] \\ &\quad \times \left[p_0^1, p_0^2, \ldots, p_0^m, p_1^1, p_1^2, \ldots, p_1^m, p_n^1, p_n^2, \ldots, p_n^m\right]^T \quad \text{i} = 1, 2, \ldots N \end{aligned} \tag{12}$$

Let $X = [x_1, x_2, .., x_N]^T$ denote the input vector, $Y = [y_1, y_2, .., y_N]^T$ denote the output vector. Meanwhile, the

Table 3 The input membership definition and output parameter values

ALT/AST	Parameter value of input variables	EI (Score)	Parameter value of input variables	The dosage strength	Parameter value of output
0	(0.01, 0.01, 1.0, 1.5)	Normal	(0, 0, 2.0, 3.5)	0	(0, 0, 0, 0)
I	(1.0, 1.5, 2.25, 2.75)	Not good	(2.0, 3.5, 6.5, 8.0)	A	(0, 0, 0.2, 0.3)
II	(2.25, 2.75, 4.5, 5.5)	Serious	(6.5, 8.0, 10, 10)	B	(0.2, 0.3, 0.45, 0.55)
III	(4.5, 5.5, 9.5, 10.5)			C	(0.45, 0.55, 0.7, 0.8)
IV	(9.5, 10.5, 15, 15)			D	(0.7, 0.8, 1, 1)

$T = [T_1, T_2, .., T_N]^T$ denote the actual output vector. By matrix transformation, we get the equation as follows:

$$\psi = \begin{bmatrix} \overline{\alpha}_1^1 \cdots \overline{\alpha}_1^m & \overline{\alpha}_1^1 x_{11} & \cdots & \overline{\alpha}_1^m x_{11} & \overline{\alpha}_1^1 x_{11} & \cdots & \overline{\alpha}_1^m x_{nN} \\ \vdots \quad \vdots \quad \vdots & \vdots & & \vdots \quad \vdots & \vdots & & \vdots \quad \vdots \\ \overline{\alpha}_N^1 \cdots \overline{\alpha}_N^m & \overline{\alpha}_N^1 x_{1N} & \cdots & \overline{\alpha}_N^m x_{1N} & \overline{\alpha}_N^1 x_{nN} & \cdots & \overline{\alpha}_N^M x_{nN} \end{bmatrix}_{N\times m(n+1)} \tag{13}$$

$$P = \left[p_0^1, p_0^2, \ldots, p_0^m, p_1^1, p_1^2, \ldots, p_1^m, p_n^1, p_n^2, \ldots, p_n^m\right]^T$$

P is consequent parameters, which needs to be estimated. The system error is $\xi = \|T-Y\|$. Find an optimal coefficient vector P* such that the system error is minimized. This problem can be solved by the linear least squares (LLS) method, implemented by approximating $P = (\psi^T\psi)^{-1}\psi^T Y$. In order to avoid calculating the inverse matrix, the kalman filter algorithm (KF) [22, 35, 47] is used to determine the P:

$$\begin{array}{l} P_{i+1} = P_i + s_{i+1}\psi_{i+1}^T\left(T_{i+1} - \psi_{i+1}P_i\right) \\ s_{i+1} = s_i - \frac{s_i\psi_{i+1}^T\psi_{i+1}s_i}{1+\psi_{i+1}s_i\psi_{i+1}^T}, i = 1, 2 \ldots, n-1 \end{array} \tag{14}$$

Initial condition is $p_0=0$, $S_0=\lambda I$. ψ_{i+1} is the i+1 row of the ψ, and P_i is the coefficient matrix after the i^{th} iteration. λ is a bigger constant usually, here is 5000, I is a $m(n+1)\times m(n+1)$ unit matrix.

The hybrid learning algorithm description for T-S FNNs

Step 1: Initialize population size, particle size and related parameters;

Step 2: Generate randomly the membership function parameters of input variables, and then, according to the Eq. 14, the consequent parameters of T-S fuzzy model are estimated;

Step 3: Code the parameters. Moreover, "the golden section" method is used to determine the dimensions of population division;

Step 4: Initialize the positions and velocities of the particles of all the sub-swarms, calculate the fitness of all the particles, and initialize the personal best position and global best position of each sub-swarm;

Step 5: Generate the execution random sequence with co-evolution, and then begin cooperative optimization for each sub-swarm;

Step 6: Satisfy the iteration period predefined by users, according to the selection criteria, the best performance particle is decomposed, and then the minimum value of the decomposed offspring replace the value of the mother particle;

Step 7: When the mutation mechanism criterion is reached, the "exponential smoothing mutation" module will be started. After mutation, the particles with good performance are chosen to replace the worse particles with the same amount;

Step 8: Generate a Q group, choose a particle in the Q group randomly, and ensure that it is not the global optimal value obtained by integrating RCDPSO/DPSO algorithm. Calculate the fitness of all the particles in evolution to find the optimal solution;

Step 9: Stop scheme: The stopping criterion is reached whenever one of the following conditions is fulfilled: (1) a given fitness value is achieved; (2) the total number of generations specified by users is reached. Once the stopping criterion is reached, go to step 10; otherwise, go to step 5;

Step 10: Decode and generate T-S model parameters, the optimal solution is output, and the algorithm is terminated.

Table 4 The fuzzy rule base of liver damage variances

Number	ALT grade	AST grade	EI	Y1	Y2	Y3
1	I	I	Normal	C	0	0
2	I	II	Not good	B	A	0
3	I	III	Normal	C	A	A
4	I	IV	Serious	C	B	B
5	II	0	Not good	B	0	0
6	II	III	Not good	C	B	A
...	...	...	...	...	...	...
73	III	III	Normal	D	C	C
74	IV	III	Not good	D	C	C
75	IV	IV	Serious	C	C	C

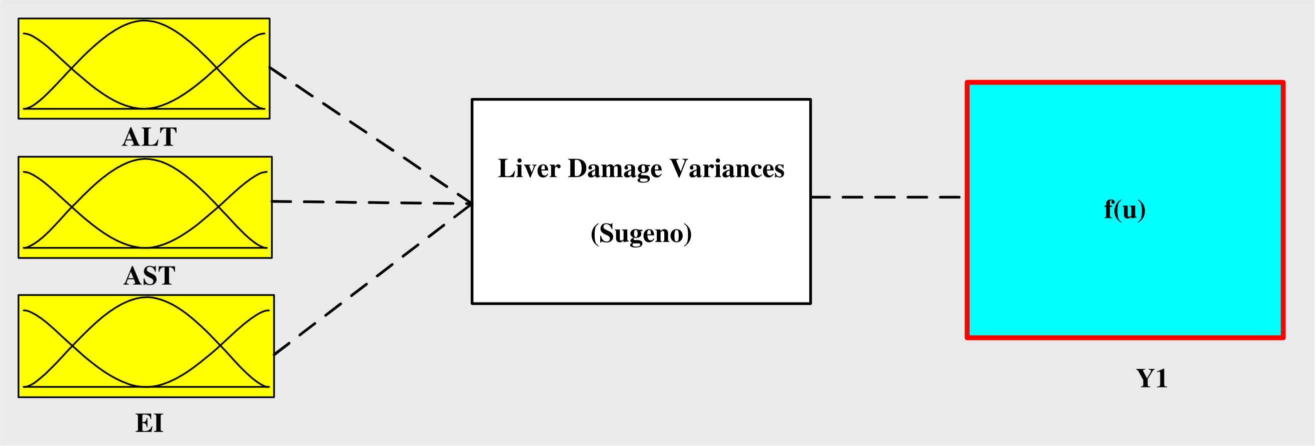


Fig. 1 Takagi-Sugeno method with three inputs and one output

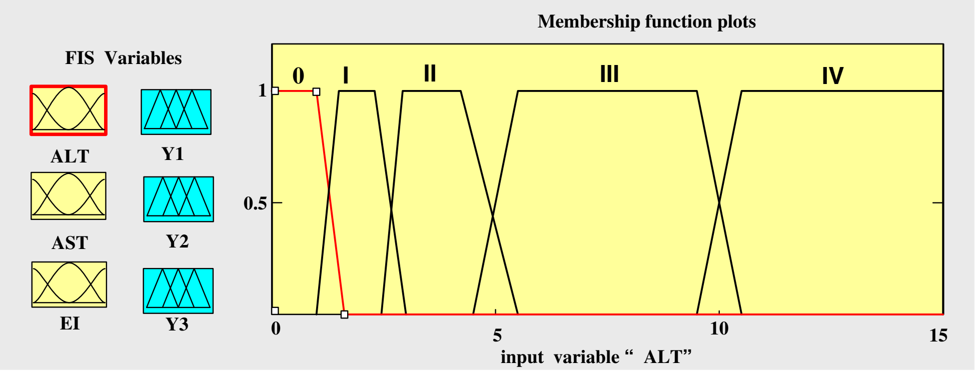
Case study

In this part, an example on the CP of osteosarcoma preoperative chemotherapy (variances usually occur in this stage) is given to validate the proposed method. The aim of this study was to improve the accuracy and efficiency of variances handling for CPs. Furthermore, the experiment can be divided into three main stages. Firstly, according to the variances characteristics of osteosarcoma preoperative chemotherapy and experts' knowledge, the membership function of evaluation index and the fuzzy rule base can be established. Secondly, according to the obtained fuzzy rules, the T-S FNNs model can be constructed. Thirdly, the constructed standard T-S FNNs and the T-S FNNs with CPSO, PSO and RCDPSO/DPSO optimization are respectively tested with the same training and test samples to compare the accuracy and efficiency in variances handling of CPs.

As usual, combining many different kinds of chemotherapy drugs is adopted as the osteosarcoma preoperative chemotherapy scheme. In this study, the preoperative chemotherapy scheme is simplified to "high-dose methotrexate (HDMTX) + calcium folinate (CF) rescue". As usual, the variances can be listed as follows: 1) Skin rash; 2) Stomatitis; 3) Gastrointestinal reactions; 4) Liver damage; 5)Renal damage; 6) Myelosuppression [5]. Moreover, during the process of preoperative chemotherapy, liver damage occurs with a higher percentage, which can directly influence the life quality of patients and further execution of the chemotherapy scheme. Therefore, in this paper, we pay more attention to liver damage variances. The evaluation of liver damage includes mandatory indexes: 1) Alanine Amino Transferase (ALT); 2) Aspartate aminotransferase (AST) and the experience index (EI). According to these indexes, the dosage of liver protection drugs can be determined. The membership functions of these indexes are listed as follows:

Index 1: Alanine Amino Transferase (ALT). ALT can be classified into 0, I, II, III, IV five grades, where 0 means that the liver function is normal, while IV means that the liver function damage is very severe. The membership function definition and parameters' values of ALT is shown in Table 3. The ULN is the upper limit of the normal status, meaning that when 0≤ the amount of ALT≤1.5×ULN, the liver function is normal; when 1×ULN≤ the amount of ALT≤2.75×ULN, the liver function is minor damage; when 2.25×ULN≤ the amount of ALT≤5.5×ULN, the liver

Fig. 2 Membership functions of input variable "ALT"



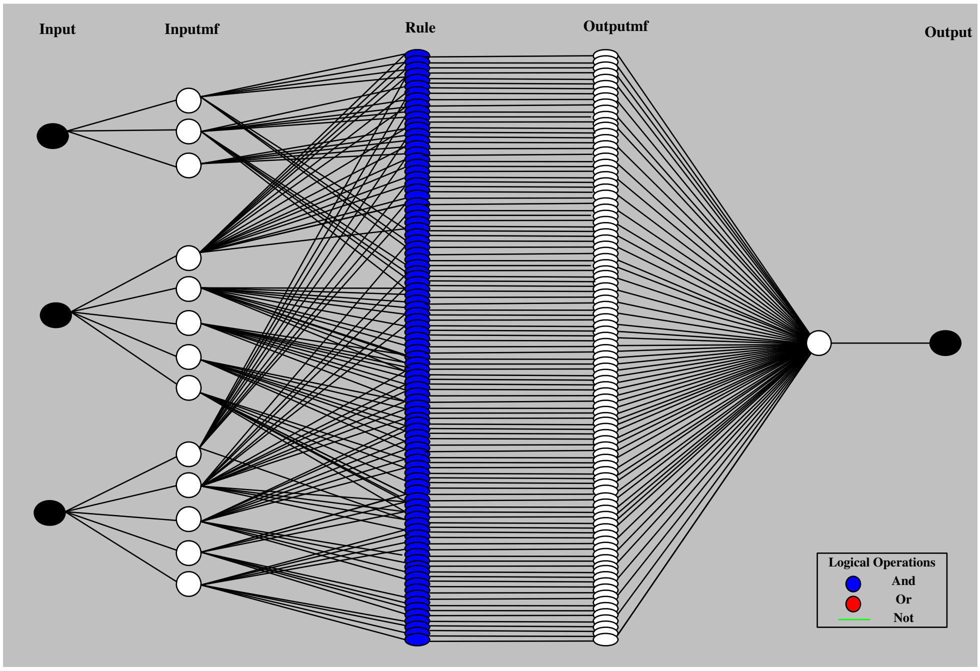


Fig. 3 ANFIS architecture

function is moderate damage; when the 4.5×ULN≤ the amount of ALT≤10.5×ULN, the liver function is severe damage; when 9.5×ULN≤ the amount of ALT≤15×ULN, the liver function is very severe damage.

The membership function of index 2 (AST) is equal to index 1(ALT). Besides the AST and ALT, the experience index (such as the patient's age, sex, and medical history) should be considered. Therefore, according to the medical expert experience, an experience index (EI) can be constructed to consider all "other factors" comprehensively. Moreover, the scoring method is adopted for the EI. It can be divided into "normal", "not good", "serious" three parts, as shown in Table 3. The trapezoidal membership function is adopted.

The output variables are the dosage strength of protecting liver drugs. When a liver poisoning happens, under normal circumstances, the liver protection drugs should be used. Based on actual situation, during the liver protection therapy, no more than three kinds of liver protection drugs should be used. Therefore, the output is set to three kinds of liver protection drugs dosage, where the three kinds of medicine will be named as "liver protection drug A", "liver protection drug B" and "liver protection drug C", followed by the symbol of the "Y1", "Y2", "Y3". The dosage of the domain value is [0, 1], where "0" means no drugs, "1" indicates that the kinds of drugs the maximum allowable dose. Fuzzy output space is divided into five parts, namely "0-class", "A-class", "B-class", "C-class" and "D-class". The range of different parameter values can be shown in Table 3.

The fuzzy partition number of the fuzzy input space determines the maximum possible number of fuzzy rules. This fuzzy system has three inputs, namely ALT, AST and EI, in which the number of fuzzy partition is respectively 5, 5 and 3, so the maximum number of rules in the system is 5×5×3=75. The format of fuzzy rules can be listed as follows:

Rule: **IF** ALT level is I grade **and** AST level is I grade **and** the experience index is normal

THEN

a strength of the liver protection drug A is C-class,
a strength of the liver protection drug B is 0-class,
a strength of the liver protection drug C is 0-class.

The fuzzy rules have three inputs and three outputs. Therefore, it belongs to multiple-input and multiple-output (MIMO) systems. The three output actions are mutually independent, which can be split into three sub-rules of MISO. Suppose the rule's reliability is 1. According to experts' knowledge of osteosarcoma, the fuzzy rule base can be established as shown in Table 4.

Experiments on T-S FNNs

T-S FNNs has three outputs: Y1, Y2 and Y3, which should be split into three MISO systems. T-S model with three inputs and one output Y1 are used as illustrated in Fig. 1. The membership functions of input variable "ALT" is shown in Fig. 2. According to these rules depicted in

Table 5 The results of T-S fuzzy model related to various alternative structures

Input function			Output function					
Function type		Fuzzy splitting	Linear			Constant		
			Training error	Test error	Difference	Training error	Test error	Difference
Y1	trapmf	3,3,3	0.0541	0.0689	−0.0148	0.0655	0.0455	0.0200
	trapmf	3,3,5	0.0518	0.1008	−0.0490	0.0635	0.0492	0.0143
	trapmf	5,5,3	0.0318	0.5987	−0.5669	0.0410	0.0318	0.0092
	trapmf	5,5,5	0.0258	0.4608	−0.4350	0.0503	0.0452	0.0051
	gaussmf	3,3,3	0.0448	0.0598	−0.0150	0.0620	0.0505	0.0115
	gaussmf	3,3,5	0.0421	0.0706	−0.0285	0.0652	0.0586	0.0066
	gaussmf	5,5,3	0.0280	0.2072	−0.1792	0.0462	0.0297	0.0165
	gaussmf	5,5,5	0.0128	0.4578	−0.4450	0.0414	0.1240	−0.0826
Y2	trapmf	3,3,3	0.0260	0.0593	−0.0333	0.0386	0.0488	−0.0102
	trapmf	3,3,5	0.0310	0.1004	−0.0694	0.0382	0.0417	−0.0035
	trapmf	5,5,3	0.0121	0.8837	−0.8716	0.0263	0.0243	0.0020
	trapmf	5,5,5	0.0127	0.5397	−0.5270	0.0252	0.0422	−0.0170
	gaussmf	3,3,3	0.0249	0.0356	−0.0107	0.0404	0.0441	−0.0037
	gaussmf	3,3,5	0.0309	0.0469	−0.0160	0.0383	0.0454	−0.0071
	gaussmf	5,5,3	0.0106	1.4184	−1.4078	0.0209	0.8321	−0.8112
	gaussmf	5,5,5	0.0046	0.7886	−0.7840	0.0267	0.2028	−0.1761
Y3	trapmf	3,3,3	0.0355	0.0853	−0.0498	0.0580	0.0712	−0.0132
	trapmf	3,3,5	0.0360	0.1021	−0.0661	0.0466	0.0572	−0.0106
	trapmf	5,5,3	0.0159	0.1853	−0.1694	0.0263	0.0737	−0.0474
	trapmf	5,5,5	0.0124	0.4100	−0.3976	0.0196	0.3760	−0.3564
	gaussmf	3,3,3	0.0320	0.0538	−0.0218	0.0537	0.0727	−0.019
	gaussmf	3,3,5	0.0258	0.1287	−0.1029	0.0550	0.0590	−0.004
	gaussmf	5,5,3	0.0108	0.7606	−0.7498	0.0263	0.0737	−0.0474
	gaussmf	5,5,5	0.0001	0.2941	−0.2940	0.0196	0.3760	−0.3564

Table 4, the ANFIS architecture is constructed in Fig. 3. ANFIS system was used at the MATLAB environment.

After the training samples and test samples are loaded, the fuzzy inference system is generated by the grid partition method, and the target error of 0.001 is set up. And the T-S fuzzy model is tested with varying numbers and types of input-output membership functions (MFs) of the model with constructed 75 rules. The input membership function is chosen for the trapezoid distribution (trapmf) or gaussian distribution (gaussmf) respectively. The fuzzy partition number is 5 or 3, the output membership function is linear (Linear) or constant (Constant).

The neural network performance criterion is measured by the mean absolute error (MAE) between the predicted outcome and the actual outcome. The collection of well-distributed, sufficient, and accurately measured input data is the basic requirement to obtain an accurate model. The 430 real samples have been collected from Shanghai No. 6 People's Hospital of China. The samples had been randomly split into a training data set containing 380 samples and a test data set containing the remaining 50 samples. The experiments were carried out by using cross-validation method on all data sets, and the testing data sets were used to verify the accuracy and the effectiveness of the trained T-S model.

The best model error of test samples as the final model, and the gauss membership function (gaussmf) with 0.0297 test error for Y1, trapezoid membership function (trapmf) with 0.0243 test error for Y2, and gaussmf with 0.0538 test error for Y3 can be obtained. The detailed results of the final model for Y1, Y2 and Y3 are shown in Table 5.

Experiments on T-S FNNs based on RCDPSO/DPSO, CPSO and PSO optimization

The experiments are carried on the RCDPSO/DPSO, CPSO and PSO-based T-S FNNs, as well as standard T-S fuzzy model respectively. The same 380 samples for training groups and 50 samples for testing groups are used. The experiments were carried out by using cross-validation method on all data sets. The initial parameters can be given in Table 6.

Each method is tested for 10 times. The mean is regarded as final result, and it can be seen in Table 7. The

Table 6 Parameters setting of RCDPSO/DPSO based T-S FNNs

Parameters	Value
Number of individuals in population	20
Generations of RCDPSO/DPSO	500
Generations of Q swarm	30
The split factor K	6
Initial position range and velocity of particles in RCDPSO	[−1.0 1.0]
Inertia weights w_{max}, w_{min} in RCDPSO	0.9, 0.2
The predefined iteration cycle for the double mutation	60
The fitness value of stopping evolution	0.0001
The length of the life period	30
Acceleration coefficients c_{1i} , c_{1f} , c_{2i} and c_{2f} in RCDPSO	2.5, 0.5,0.5,2.5
Limitation range of velocity w_{max}, w_{min} in Q swarm	[1.0, −1.0]

comparison learning curves with MSE and MAE of the T-S fuzzy model based on the RCDPSO/DPSO, CPSO and PSO is shown in Fig. 4.

The prediction errors and deviation comparison diagrams of the target output and actual output with 50 groups tested samples are given from Figs. 5 and 6.

As shown in Fig. 4, after the RCDPSO/DPSO-based T-S fuzzy model is trained for 500 times, the training samples of the MSE can reach 0.00051, MAE can reach 0.01352, less than 2%. The MSE for the test samples is 0.0013, MAE is 0.0166, showing good generalization performance. During the training process, the declining velocity of the error is very fast, the MAE gets to <0.04 for 30 times, and then it reaches a stable phase. Meanwhile, the MSE for the training reaches a stable phase for 300 times. However, the training error of MAE with the PSO-based T-S fuzzy model is still 0.08 for 30 times, and it get to 0.0325 for 500 times. Its prediction accuracy is less than the same input conditions of the RCDPSO/DPSO and CPSO model. By analysis, the test sample error of the RCDPSO/DPSO-based T-S method is relatively small. However, there is a relatively large error range for PSO method. For T-S fuzzy model, the mean errors of Y1, Y2 and Y3 of the training samples and test samples are 0.0348 and 0.0359 respectively. The MSE and MAE are all slightly higher than the PSO-based T-S model with 0.0325 and 0.0356. The change comparison diagram of membership function (MF) with is shown in Fig. 7. Finally, after the optimal structure and parameters have been achieved by integrating RCDPSO/DPSO-based T-S fuzzy neural network, three rules abstracted from the RCDPSO/DPSO based T-S fuzzy model are listed as follows:

Rule 1: IF x_1 is μ(0.01, 0.01, 1.0, 1.5), x_2 is μ(1.0,1.5,2.25,2.75), x_3 is μ(0, 0, 2, 3.5)

THEN
$Y=0.3109\ x_1+0.5264\ x_2+0.0213x_3-0.6728$

Rule 2: IF x_1 is μ(1.0,1.5,2.25,2.75), x_2 is μ(1.0,1.5,2.25,2.75), x_3 is μ(2,3.5,6.5,8)

THEN
$Y=0.2484\ x_1+0.1671\ x_2+0.0023x_3+0.0380$

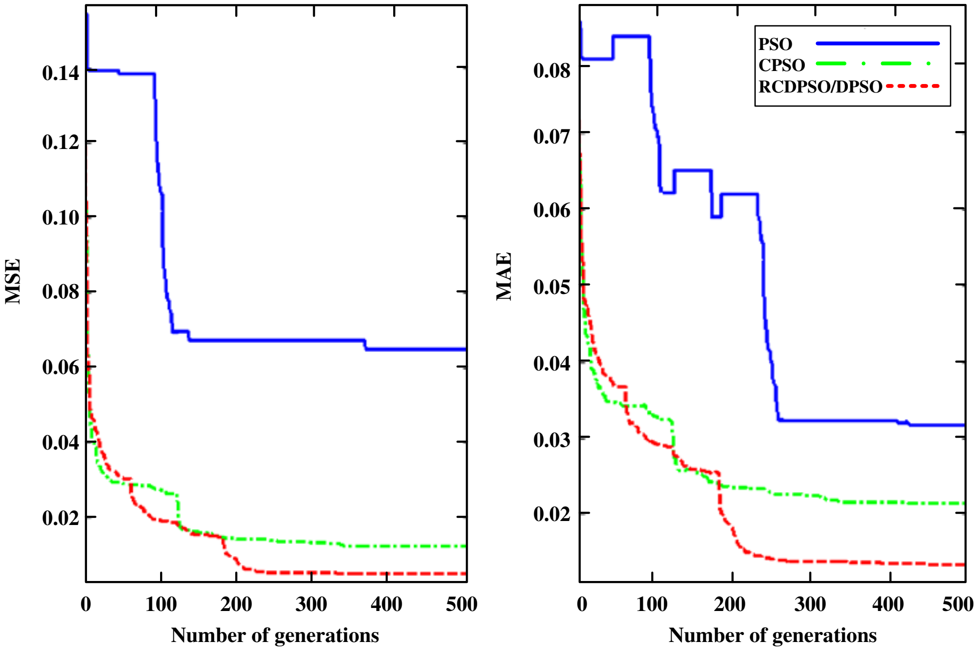


Fig. 4 The comparison learning curves of T-S model based on RCDPSO/DPSO, CPSO and PSO

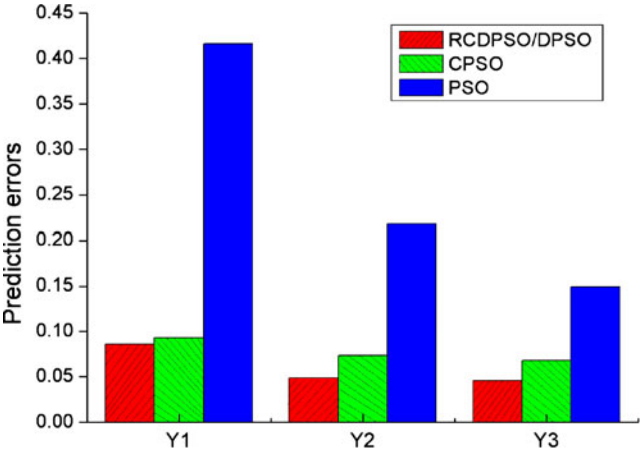


Fig. 5 The prediction errors of T-S model based on RCDPSO/DPSO, CPSO and PSO for Y1, Y2 and Y3

Rule 3: IF x_1 is μ(4.5,5.5,9.5,10.5), x_2 is μ(1.0,1.5,2.25,2.75), x_3 is μ(2,3.5,6.5,8)

THEN $Y=-0.1015\ x_1+0.0318\ x_2+0.0156x_3+0.3951$

Where μ(a,b,c,d) is the linguistic value corresponding to the membership function.

The experimental comparisons analysis of T-S FNNs based on RCDPSO/DPSO, CPSO and PSO optimization

In order to further compare and analyze the overall performance of T-S FNNs based on RCDPSO/DPSO, CPSO and PSO optimization with the optimal solution (the actual value), the same 430 samples are experimented. In one trial, a certain number of samples, denoted as training-size, are randomly selected from the data set as the training samples. And 40 samples are chosen for testing samples. The remaining samples are used as testing samples. Each neural network is then trained and tested 10 times. Its average result is recorded as the final result. In this research, the size of the problems (example size) varied over n=30, 60,..., 390. That is to say, we run the trials over the network with training-size ranging from 30 to 390.

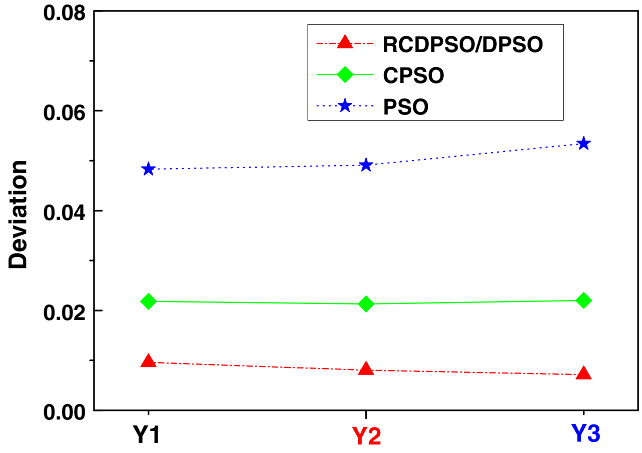


Fig. 6 The deviation comparison of T-S model based on RCDPSO/DPSO, CPSO and PSO for Y1, Y2 and Y3

From largest to smallest they are: T-S FNNs based on PSO, CPSO and RCDPSO/DPSO optimization. In this study, the optimal solution is known (the actual value). According to literature [48], the metric chosen is (Z-Z*)/Z* (where Z is the heuristic solution value and Z* is the actual value), which expresses the result as a proportion of the optimal solution.

Figure 8 shows the mean value of this metric for each heuristic, again as a function of problem size n. It can be seen that for all heuristics, the metric increases nonlinearly with n.

Figure 9 shows the means of the log-transformed metric values. As with the previous figures, it appears that T-S FNNs RCDPSO/DPSO optimization outperforms T-S FNNs based on CPSO optimization, which in turn outperforms T-S FNNs based on PSO optimization for all n.

All three heuristics' metrics seem to be different and seem to depend upon n.

From Figs. 8 and 9, it is obvious that the deviation of T-S FNNs based on RCDPSO/DPSO optimization is the smallest across different training-sizes, which means that the T-S FNNs based on RCDPSO/DPSO optimization network is more stable and robust than T-S FNNs based on CPSO and PSO optimization regardless of the training-size. Therefore, the T-S FNNs based on RCDPSO/DPSO optimization can obtain a relative high accuracy to provide an effective decision support tool for fuzzy and uncertainty variances handling of CPs.

The comparison of experimental results with other methods

In order to verify the effectiveness of FNNs in handling variances of CPs, the standard Mamdani fuzzy neural network is also tested with the same training and test data. It can be divided into the following three scenarios for experiments analysis:

(1) The membership function of input variables is known in advance, and training FNNs is used to adjust the "weight" to get the minimum error. This method can be labeled as Mamdani_km;
(2) The fuzzy segmentation and membership function of input variables is determined according to expert's experience. However, the parameters of membership function can only be estimated approximately, and all these parameters and weight in fuzzy neural network

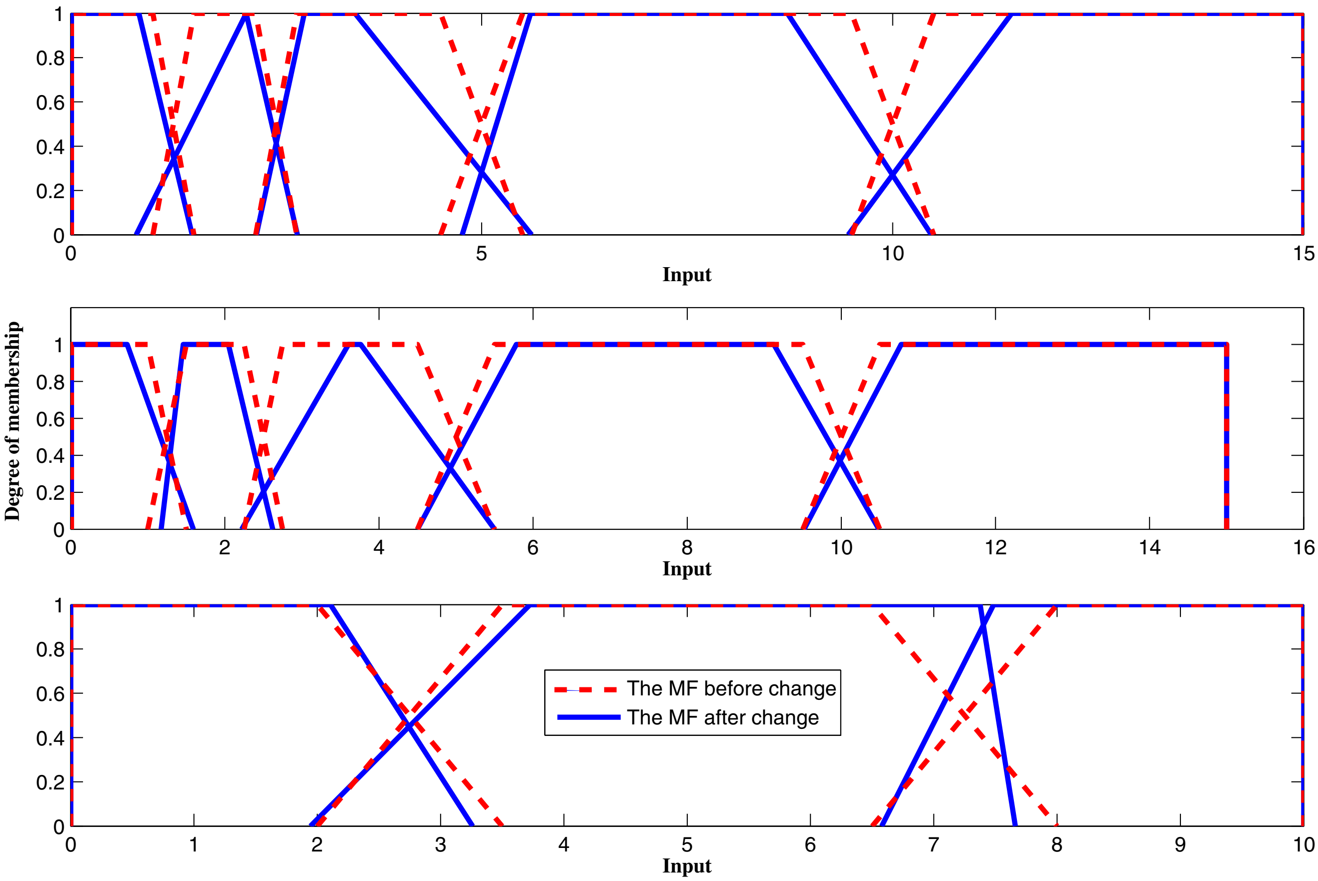


Fig. 7 The change comparison diagram of membership function

training process needed to be adjusted. This method can be labeled as Mamdani_am;

(3) The fuzzy segmentation and membership function of input variables is known, however, the parameters of fuzzy membership function are completely unknown, and they will be generated randomly. This method can be labeled as Mamdani.

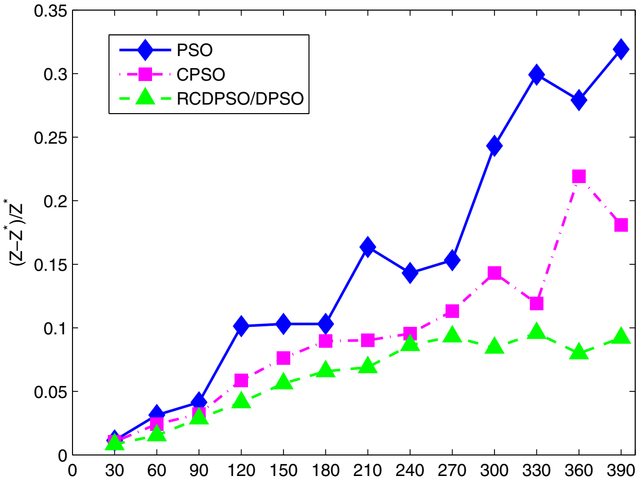


Fig. 8 Means of (Z-Z*)/Z*

Similarly, with the same parameters conditions for the RCDPSO/DPSO-based T-S fuzzy model, they can be labeled as the RCDPSO/DPSO_km, RCDPSO/DPSO_am and RCDPSO/DPSO respectively. The results and segmental comparison diagram are shown in Table 7, and Fig. 10. The test error of "Mamdani" fuzzy model is 0.3098, and the value is very big. Therefore, it can not be shown in Fig. 10.

Experimental results can be summarized as follows:

- When the input variables all are random or approximate initial parameters, the mean square error (MSE) and the mean absolute error (MAE) of the RCDPSO/DPSO-based T-S FNNs model are the smallest. The error of the CPSO-based T-S FNNs is lower than the RCDPSO/DPSO-based one, however it is higher than the PSO-based T-S FNNs and the standard T-S FNNs model;
- When the input of the membership function is known, and is determined according to expert's experience, the RCDPSO/DPSO-based T-S fuzzy model can get the best result. However, it will take a long time. Therefore, the standard Mamdani or T-S FNNs model is another good choice, because this neural network can make full use of existing expert knowledge, which can significantly reduce the uncertainty of the neural network.

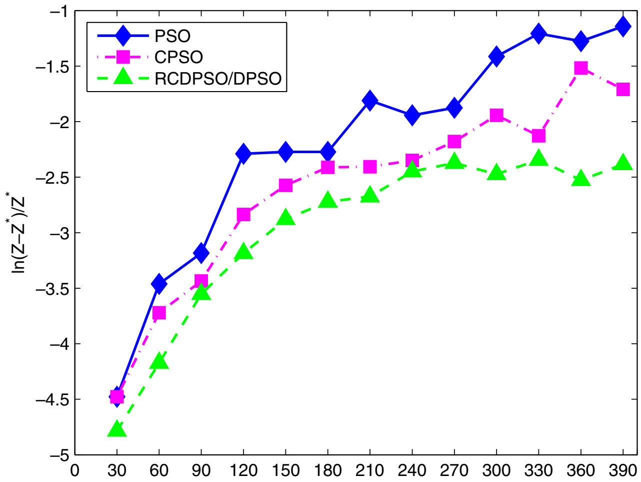


Fig. 9 Means of ln (Z-Z*)/Z*

All these indicate that the proposed RCDPSO/DPSO-based T-S FNNs is better in efficiency and precision for variances handling of CPs. Although the RCDPSO/DPSO_km-based T-S FNNs provides smaller MSE and MAE, the initial parameters of membership function are often unknown, which make it infeasible in a real-life situation.

The comparison of diagnosis accuracy rate and discussion

As is known to us all, the liver is an important metabolism and detoxification organ, which participates in the metabolism of protein, fat, glucose metabolism and drugs detoxification. Especially, during osteosarcoma preoperative chemotherapy, the dosage strength and kinds of protecting liver drugs are very important to use for the patients with liver damage variances. And any excess or unfit drugs can increase liver burden and even damage liver function. The reason of causing liver damage with unsuitable drugs can be listed as follows:

(1) Some drugs can direct damage the liver before hepatic metabolism;
(2) Some drugs can damage the liver during hepatic metabolism;
(3) During osteosarcoma preoperative chemotherapy, any drug is sensitive to patients.

Therefore, for liver poisoning variances in osteosarcoma preoperative chemotherapy, too many kinds and excessive doses of liver protection drugs can aggravate liver burden, and even cause complicating disease. The proper using liver protection drugs not only can cure patients' liver poisoning variances but also can reduce medical expense of osteosarcoma CP.

In this paper, to validate the proposed method, the accuracy of using liver protection drugs can be determined by the computation of the diagnosis accuracy rate.

$$\text{The diagnosis accuracy rate} = \frac{\text{Number of correct decisions}}{\text{Total number of samples}} \times 100\%$$

Here, the correct decisions means that the values of prediction result and actual result belong to the same "class". For example, the actual value of the dosage strength for protecting liver drugs with patient "Jack" is

Table 7 The performance comparisons of different models

Method		MSE			MAE		
		Training error	Test error	Difference	Training error	Test error	Difference
RCDPSO/DPSO	mean	0.00051	0.0013	−0.00079	0.01352	0.0166	−0.00308
	max	0.0006	0.0014	−0.0008	0.0139	0.0173	−0.0034
	min	0.0004	0.0011	−0.0007	0.0125	0.0162	−0.0037
CPSO	mean	0.0014	0.0043	−0.0029	0.0240	0.0409	−0.0169
PSO	mean	0.0062	0.0078	−0.0020	0.0325	0.0356	−0.0031
RCDPSO/DPSO_km	mean	0.00037	0.0011	−0.00073	0.0010	0.0113	−0.0103
	max	0.00039	0.0013	−0.00091	0.0014	0.0118	−0.0104
	min	0.00035	0.0010	−0.00065	0.0009	0.0109	−0.0100
RCDPSO/DPSO_am	mean	0.00045	0.0013	−0.00085	0.0117	0.0167	−0.005
Mamdani	mean	0.0073	0.3098	−0.3025	0.0591	0.1408	−0.0817
Mamdani_km	mean	0.0017	0.0026	−0.0009	0.0184	0.0198	0.0075
Mamdani_am	mean	0.0035	0.0046	−0.0010	0.0392	0.0454	−0.0062
T-S_Y1	–	–	–	–	0.0462	0.0297	0.0165
T-S_Y2	–	–	–	–	0.0263	0.0243	0.0020
T-S_Y3	–	–	–	–	0.0320	0.0538	−0.0218

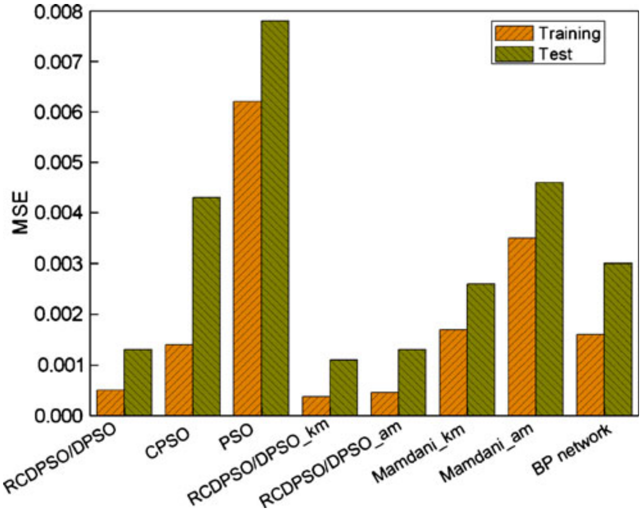


Fig. 10 The segmental comparison results of different algorithms

"A" class. After adopting the proposed method for prediction, the prediction value of the dosage strength for protecting liver drugs with patient "Jack" is still "A" class. By this criterion, the network model is then tested 10 trials with the given test-set-size. The same 50 testing samples are used. The final accuracy rate with the test-set-size (10, 20 and 50) is determined by the average accuracy rate for the testing samples in 10 trials. Table 8 gives the obtained diagnosis accuracy rates with different train-set-sizes. From Table 8, the different models with the average diagnosis accuracy rate is 98.19%, 95.33%, 84.67%, 99.84%, 99.16%, 81.20%, 94.24%, 88.72%, 82.95%, and 81.57% respectively. The diagnosis accuracy rate of the RCDPSO/DPSO-based T-S fuzzy model is 98%. Meanwhile, the diagnosis accuracy rate of the standard T-S fuzzy model (the average value of Y1, Y2 and Y3) is 82.95%.

Especially, the proposed RCDPSO/DPSO-based T-S fuzzy model or RCDPSO/DPSO_km -based T-S fuzzy model outperforms above 15% accuracy of the standard T-S fuzzy model or the standard Mamdani fuzzy model. Moreover, the T-S FNNs with PSO or improved PSO optimization can achieve better results than the standard T-S FNNs and the standard BP neural network for the variances handling of CPs. As usual, the input variables are random or approximate initial parameters, the proposed RCDPSO/DPSO-based T-S FNNs model is often needed, although the RCDPSO/DPSO_km-based T-S FNNs can obtain higher diagnosis accuracy rate than the proposed RCDPSO/DPSO-based T-S FNNs model. Thus, these results indicate that the proposed RCDPSO/DPSO-based T-S FNNs model is effective in handling CPs variances.

Conclusion

In this paper, a new variances handling method of CPs is proposed, which is based on T-S FNNs with integrating RCDPSO/DPSO optimization. The parameters optimization consists of two phases: first, all structure and antecedent parameters are adjusted by integrating RCDPSO/DPSO optimization; second, all consequent parameters are estimated through kalman filtering algorithm. During the process of cooperative evolution with random execution sequence, a decomposing algorithm is adopted for the particles with the best performance. Therefore, it not only ensures the convergence rate, but also improves performance of the algorithm in global search. Moreover, the exponential smoothing mutation mechanism is adopted to further improve the solutions found by the RCDPSO optimization. Therefore, the optimal structure and parameters can be achieved simultaneously by novel hybrid learning algorithm.

To demonstrate the performance of the proposed method, a case study on liver poisoning variances problems of osteosarcoma preoperative chemotherapy is tested. The results demonstrate that the RCDPSO/DPSO-based T-S

Table 8 The obtained diagnosis accuracy rates of different methods

Method	Diagnosis accuracy rate			
	10 Test-set-size	20 Test-set-size	50 Test-set-size	Average
RCDPSO/DPSO	97.92%	98.14%	98.5%	98.19%
CPSO	95.23%	95.38%	95.38%	95.33%
PSO	82.22%	84.46%	87.34%	84.67%
RCDPSO/DPSO_km	99.83%	99.84%	99.85%	99.84%
RCDPSO/DPSO_am	99.12%	99.16%	99.20%	99.16%
Mamdani	77.85%	81.89%	83.87%	81.20%
Mamdani_km	94.20%	94.27%	94.25%	94.24%
Mamdani_am	88.74%	88.71%	88.71%	88.72%
T-S FNNs	79.58%	82.56%	86.72%	82.95%

FNNs can obtain better performance with high accuracy, efficiency and generalization ability than other existing methods. Moreover, based on the trained RCDPSO/DPSO-based T-S FNNs, the interrelated rules would be extracted, which can be used to provide doctors or healthcare professionals with the guidelines for management and control of complex CPs variances. Moreover, this method is also applicable to other CPs variances (e.g. cesarean section CPs hemorrhage variances). Thus, it verifies that the RCDPSO/DPSO-based T-S FNNs is an effective decision support tool for fuzzy and uncertainty variances handling of CPs.

Acknowledgement This work described in this paper was supported by Research Grant from National Natural Science Foundation of China (60774103) and Major Program Development Fund of SJTU. Moreover, we would also like to thank to the whole medical staff of Shanghai No. 6 People's Hospital for real data collecting and helpful discussions.

The authors would like to express sincere appreciation to the journal editor and two anonymous referees for their detailed and helpful comments to improve the quality of the paper.

References

1. Bragato, L., and Jacobs, K., Care pathways: the road to better health services. *J. Health Organ. Manage.* 17(3):164–180, 2003.
2. Cheah, J., Development and implementation of a clinical pathway programme in an acute care general hospital in Singapore. *Int. J. Qual. Health Care* 12:403–412, 2000.
3. Hunter, B., and Segrott, J., Re-mapping client journeys and professional identities: a review of the literature on clinical pathways. *Int. J. Nurs. Stud.* 45(4):608–625, 2008.
4. Wakamiya, S. J., and Yamauchi, K., What are the standard functions of electronic clinical pathways? *Int. J. Med. Inform.* 78 (8):543–550, 2009.
5. Du, G., Jiang, Z. B., Diao, X. D., Ye, Y., Yao, Y., Modeling, variation monitoring, analyzing, reasoning for intelligently reconfigurable clinical pathway. Proceedings of the IEEE International Conference on Service Operations, Logistics and Informatics, 85–90, Chicago, IL, USA, 2009.
6. Wigfield, A., and Boon, E., Critical care pathway development: the way forward. *Br. J. Nurs.* 5(12):732–735, 1996.
7. Cheah, J., Clinical pathways: changing the face of client care delivery in the next millennium. *Clinician Manag.* 7(78):78–84, 1998.
8. Price, M. B., Jones, A., Hawkins, J. A., et al., Critical pathways for postoperative care after simple congenital heart surgery. *AJMC* 5:185–192, 1999.
9. Atwal, A., and Caldwell, K., Do multidisciplinary integrated care pathways improve interprofessional collaboration? *Scand. J. Caring Sci.* 16(4):360–367, 2002.
10. Bryan, S., Holmes, S., Prostlethwaite, D., and Carty, N., The role of integrated care pathways in improving the client experience. *Prof. Nurse* 18(2):77–79, 2002.
11. Caminiti, C., Scoditti, U., Diodati, F., and Passalacqua, R., How to promote, improve and test adherence to scientific evidence in clinical practice. *BMC Health Serv. Res.* 5(62):1–11, 2005.
12. Wakamiya, S., and Yamauchi, K., A new approach to systematization of the management of paper-based clinical pathways. *Comput. Meth. Programs Biomed.* 82:169–176, 2006.
13. Ye, Y., Jiang, Z. B., Diao, X. D., Yang, D., and Du, G., An ontology-based hierarchical semantic modeling approach to clinical pathway workflows. *Comput. Biol. Med.* 39(8):722–732, 2009.
14. Ye, Y., Jiang, Z. B., Diao, X. D., Du, G., Knowledge-based hybrid variance handling for patient care workflows based on clinical pathways. Proceedings of the IEEE International Conference on Service Operations, Logistics and Informatics, 13–18, Chicago, IL, USA, 2009.
15. Ye, Y., Jiang, Z. B., Diao, X. D., Du, G., A Semantics-Based Clinical Pathway Workflow and Variance Management Framework, 2008 IEEE International Conference on Service Operations and Logistics, and Informatics, 758–762, Bei Jing, China, 2008.
16. Du, G., Jiang, Z. B., Diao, X. D., Sun, Y. J., Ye, Y., and Yao, Y., Adaptive workflow engine based on rule for clinical pathway. *Journal of Shanghai Jiao Tong University* 43 (7):1021–1026, 2009.
17. Er, O., Yumusak, N., and Temurtas, F., Chest diseases diagnosis using artificial neural networks. *Expert Syst. Appl.* 37(12):7648–7655, 2010.
18. Chen, Y., Yang, B., Abraham, A., and Peng, L., Automatic design of hierarchical Takagi–Sugeno fuzzy systems using evolutionary algorithms. *IEEE Trans. Fuzzy Syst.* 15(3):385–397, 2007.
19. Yoon, Y., Guimaraes, T., and Swales, G., Integrating artificial neural networks with rule-based expert systems. *Decis. Support Syst.* 11(5):497–507, 1994.
20. Takagi, T., and Sugeno, M., Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybern.* 15:116–132, 1985.
21. Lin, F. J., Lin, C. H., and Shen, P. H., Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive. *IEEE Trans. Fuzzy Sys.* 9(5):751–759, 2001.
22. Jang, J. S. R., ANFIS: Adaptive network based fuzzy inference system. *IEEE Trans. Syst. Man Cybern.* 23(3):665–684, 1993.
23. Pedrycz, W., and Reformat, M., Evolutionary fuzzy modeling. *IEEE Trans. Fuzzy Syst.* 11(5):652–665, 2003.
24. Oh, S. K., Pedrycz, W., and Park, H. S., Hybrid id entification in fuzzy neural networks. *Fuzzy Sets Syst.* 138:399–426, 2003.
25. Wang, L., and Yen, J., Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter. *Fuzzy Sets Syst.* 101:353–362, 1999.
26. Wang, H., Kwong, S., Jin, Y., Wei, W., and Man, K. F., Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction. *Fuzzy Sets Syst.* 149(1):149–186, 2005.
27. Tang, A. M., Quek, C., and Ng, G. S., GA-TSKfnn: parameters tuning of fuzzy neural network using genetic algorithms. *Expert Syst. Appl.* 29:769–781, 2005.
28. Lin, C. J., and Xu, Y. J., A self-adaptive neural fuzzy network with group-based symbiotic evolution and its prediction applications. *Fuzzy Sets Syst.* 157:1036–1056, 2006.
29. Jelodar, M. S., Kamal, M., Fakhraie, S. M., Ahmadabadi, M. N., SOPC-based parallel genetic algorithm. Proceedings of the IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, 2006.
30. Kennedy, J., Eberhart, R C., Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*. Piscataway, 1942–1948, 1995.
31. Fourie, P. C., and Groenwold, A. A., The particle swarm optimization algorithm in size and shape. *Struct. Multidiscip. O.* 23(4):259–267, 2002.
32. Han, M., Sun, Y. N., and Fan, Y. N., An improved fuzzy neural network based on T-S model. *Expert Syst. Appl.* 34:2905–2920, 2008.
33. Khosla, A., Kumar, S., Aggarwal, K. K., A framework for identification of fuzzy models through particle swarm optimiza-

tion, Proceedings of the IEEE Indicon Conference, 388–391, Chennai, India, 2005.

34. Khosla, A., Kumar, S., Ghosh, K. R., A comparison of computational efforts between particle swarm optimization and genetic algorithm for identification of fuzzy models, in: Annual Conference of the North American Fuzzy Information Processing Society-NAFIPS, 245–250, 2007.
35. Shoorehdeli, M. A., Teshnehlab, M., and Sedigh, A. K., Training ANFIS as an identifier with intelligent hybrid stable learning algorithm based on particle swarm optimization and extended Kalman filter. *Expert Syst. Appl.* 160:922–948, 2009.
36. Shi, Y., and Eberhart, R. C., Empirical study of particle swarm optimization. *Proc. IEEE Int. Conf. Evolutionary Computation* 3:101–106, 1999.
37. Ratnaweera, A., Halgamuge, S. K., and Watoson, H. C., Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficient. *IEEE Trans. Evol. Comput.* 8 (3):240–255, 2004.
38. Lovbjerg, M., Rasmussen, T. K., Krink, T., Hybrid particle swarm optimizer with breeding and sub-populations. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. San Francisco, CA, July 2001.
39. Xie, X. F., Zhang, W. J., et al., Optimizing semiconductor devices by self-organizing particle swarm. *Congress on Evolutionary Computation, Oregon*, USA, 2017–2022, 2004.
40. He, R., Wang, Y., et al., An improved particle swarm optimization based on self-adaptive escape velocity. *JSW* 16(12):2036–2044, 2005.
41. Lin, C.-J., Wang, J.-G., and Lee, C.-Y., Pattern recognition using neural-fuzzy networks based on improved particle swam optimization. *Expert Syst. Appl.* 36:5402–5410, 2009.
42. Lin, C.-J., An efficient immune-based symbiotic particle swarm optimization learning algorithm for TSK-type neuro-fuzzy networks design. *Fuzzy Sets Syst.* 159:2890–2909, 2008.
43. Van den Bergh, F., and Engelbrecht, A. P., A cooperative approach to particle swarm optimization. *IEEE Trans. Evol. Comput.* 8(3):225–239, 2004.
44. Niu, B., Zhu, Y. L., He, X. X., and Shena, H., A multi-swarm optimizer based fuzzy modeling approach for dynamic systems processing. *Neurocomputing* 71:1436–1448, 2008.
45. Higashi, N., Iba, H., Particle swarm optimization with Gaussian mutation. Proceedings of the IEEE Swarm Intelligence Symp. Indianapolis: IEEE Inc, 72–79, 2003.
46. Kennedy, J., Eberhart, R. C., A discrete binary version of the particle swarm algorithm, systems, man, and cybernetics, Computational Cybernetics and Simulation IEEE International Conference, 5 (12–15): 4104–4108, 1997.
47. Simon, D., Training fuzzy systems with the extended Kalman filter. *Fuzzy Sets Syst.* 132:189–199, 2002.
48. Coffin, M., and Saltzman, Statistical analysis of computational tests of algorithms and heuristics. *INFORMS J. Comput.* 12(1):24–44, 2000.