Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 Vol I,
IMECS 2010, March 17 - 19, 2010, Hong Kong

Species-based Hybrid of Electromagnetism-like Mechanism and Back-propagation Algorithms for An Interval Type-2 Fuzzy System Design

Ching-Hung Lee, *Member, IAENG*, Feng-Yu Chang, and Chung-Ta Lee

***Abstract*—In this paper, we propose a species-based hybrid of electromagnetism-like mechanism and back-propagation algorithms (SEMBP) for an interval type-2 fuzzy neural system with asymmetric membership functions (aIT2FNS). The proposed SEMBP combines the advantages of EM and BP algorithms to obtain the faster convergence and lower computational complexity. In addition, SEMBP uses the uniform method to have the initial solution agents scatter over the feasible solution region evenly and the notion of species which can locate multiple optima to provide bigger possibility of finding the global optimum. The proposed aIT2FNS system uses type-2 asymmetric fuzzy membership functions and the TSK type consequence part. Finally, the chaotic system identification problem is presented to show the performance and effectiveness of the proposed aIT2FNS with SEMBP algorithm.**

***Index Terms*—uniform initialization, asymmetric, type-2 fuzzy neural system, Takagi–Sugeno–Kang, fuzzy logic system.**

I. Introduction

Recently, it has been shown that the fuzzy neural network (FNN) which provides the advantages of both neural network and fuzzy system is successfully applied to nonlinear system identification and control [1]. In the FNN, symmetric and fixed membership functions are commonly used to simplify the design procedure. Therefore, a large number of rules should be used to achieve the specified approximation accuracy [2]. In [3], the asymmetric fuzzy membership function (AFMF) has been discussed and analyzed that it can effectively improve approximation accuracy and reduce the fuzzy rules. In the meantime, type-2 fuzzy logic system (T2FLS) has attracted more attention in many literatures [4, 5]. In [3], the interval type-2 fuzzy neural network with asymmetric membership function (IT2FNN-A) was proposed to improve the system performance. In this paper, we propose an interval type-2 fuzzy neural system with asymmetric membership function (aIT2FNS) which is the modification of IT2FNN-A [3].

In the training of the neural networks and fuzzy systems, back-propagation (BP) algorithm is widely used and is a powerful training technique [6]. Although BP can obtain the local minimum rapidly, it can not ensure finding the global solution. The optimization algorithms such as genetic algorithm, particle swarm optimization, and electromagnetism-like mechanism (EM) have less chance to get stuck in the local minimum than the gradient-based algorithms [7-9]. Literature [10] proposed an improved electromagnetism-like mechanism algorithm with back-propagation technique (IEMBP). The major modification of IEMBP is that the neighborhood random local search of EM algorithm is replaced by BP and the competitive selection concept is adopted.

In order to get better performance, we propose a species-based hybrid algorithm SEMBP where many modifications have been done to improve IEMBP. First, initial solution candidates are generated by uniform method to avoid the necessity of statistic analysis. Therefore, the solution agents scatter over the feasible solution region evenly. In notion of species, the population can be dynamically divided into subpopulations (or sub-species) based on similarity and locating multiple optima so that the bigger possibility of finding the global optimum is provided. Obviously, the SEMBP algorithm which combines the advantages of uniform method and species method improves the IEMBP. Thus, SEMBP has the advantages of faster convergence, less computation complexity and global optimization. In this paper, the proposed SEMBP algorithm is applied to train aIT2FNS. The simulation result is shown to illustrate the effectiveness of the aIT2FNS system with SEMBP algorithm.

This paper is organized as follow. Section II introduces the aIT2FNS. The proposed SEMBP is described in Section III. In Section IV, the simulation result of the chaotic system identification is presented to show the performance and effectiveness. Finally, the conclusion is given.

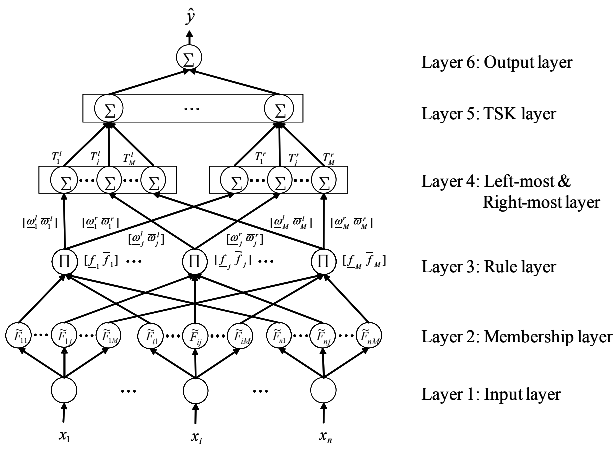


Figure 1: Diagram of MISO interval type-2 fuzzy neural system with asymmetric membership function.

II. Interval Type-2 Fuzzy Neural System with Asymmetric Membership Function

In the section, the structure of aIT2FNS is introduced. The multi-input-single-output (MISO) case is considered here for

This work was supported in part by the National Science Council, Taiwan, R.O.C., under contracts NSC-99-2221-E-155-033-MY3.
Ching-Hung Lee is with Department of Electrical Engineering, Yuan-Ze University, Chung-li, Taoyuan 320, Taiwan. (phone: +886-3-4638800, ext: 7119; fax: +886-3-4639355, e-mail: chlee@saturn.yzu.edu.tw).

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

convenience. In general, given a system input data x_i, i=1, 2, …, n and the desired output y, the jth rule is expressed as:

Rule j: IF x_1 is $\tilde{F}_{1j}$ and … and x_n is $\tilde{F}_{nj}$,

THEN $Y_j = C_{j0} + C_{j1}x_1 + C_{j2}x_2 + \cdots + C_{jn}x_n$,

where j=1, 2, …, M, C_{j0}, and C_{ji} are consequent type-1 fuzzy sets, Y_j is the output of jth rule and also a type-1 fuzzy set (a linear combination of type-1 fuzzy sets) and $\tilde{F}_{ij}$ is type-2 antecedent fuzzy set. Herein, the antecedent part of the fuzzy MFs $\tilde{F}_{ij}$ is asymmetric interval type-2 fuzzy sets which are different from typical Gaussian MFs.

The proposed aIT2FNS is implemented as the six-layer network shown in Fig. 1. We first indicate the signal propagation and the operation functions of the nodes in each layer. In the following description, $O_i^{(l)}$ denotes the ith output of a node in the lth layer.

Layer 1: Input Layer

For the ith node of layer 1, the net input and the net output are represented as:

$$O_i^{(1)} = x_i , \tag{1}$$

where i=1, 2, …, n, and x_i represents the ith input to the ith node of layer 1. The nodes in this layer only transmit input values to the next layer directly.

Layer 2: Membership Layer

In this layer, each node performs type-2 AFMF [3], i.e.,

$$O_{ij}^{(2)} = \mu_{\tilde{F}_{ij}}(O_i^{(1)})$$

$$= \left[\underline{O}_{ij}^{(2)} \quad \overline{O}_{ij}^{(2)}\right]^T = \left[\underline{\mu}_{\tilde{F}_{ij}}(O_i^{(1)}) \quad \overline{\mu}_{\tilde{F}_{ij}}(O_i^{(1)})\right]^T \tag{2}$$

where the subscript $_{ij}$ indicates the jth term of the ith input $O_i^{(1)}$, where $j = 1,\ldots,M$.

Layer 3: Rule Layer

The links in this layer are used to implement the antecedent matching and they are equal to the work in rule layer. Using the product t-norm, the firing strength associated with the jth rule is

$$\underline{f}_j = \underline{\mu}_{\tilde{F}_{1j}}(O_1^{(1)}) * \cdots * \underline{\mu}_{\tilde{F}_{nj}}(O_n^{(1)}) , \tag{3}$$

$$\overline{f}_j = \overline{\mu}_{\tilde{F}_{1j}}(O_1^{(1)}) * \cdots * \overline{\mu}_{\tilde{F}_{nj}}(O_n^{(1)}) , \tag{4}$$

where $\underline{\mu}_{\tilde{F}_{ij}}(\cdot)$ and $\overline{\mu}_{\tilde{F}_{ij}}(\cdot)$ are the lower and upper membership grades of $\mu_{\tilde{F}}(\cdot)$. Therefore, a simple PRODUCT operation is used. For the jth output rule node

$$O_j^{(3)} = \left[\underline{O}_j^{(3)} \quad \overline{O}_j^{(3)}\right]^T = \left[\prod_{i=1}^{n} \omega_{ij}^{(3)} \underline{O}_{ij}^{(2)} \quad \prod_{i=1}^{n} \omega_{ij}^{(3)} \overline{O}_{ij}^{(2)}\right]^T \tag{5}$$

where the weights $\omega_{ij}^{(3)}$ are assumed to be unity.

Layer 4: Left-most & Right-most layer

Without loss of generality, the consequent part of aIT2FNS are $\left[\underline{\omega}_j^l \ \overline{\omega}_j^l\right]^T$ and $\left[\underline{\omega}_j^r \ \overline{\omega}_j^r\right]^T$, where $\underline{\omega}_j^l < \overline{\omega}_j^l$ and $\underline{\omega}_j^r < \overline{\omega}_j^r$. The following vector notations $\underline{\omega}^l = [\underline{\omega}_1^l \cdots \underline{\omega}_M^l]^T$, $\overline{\omega}^l = [\overline{\omega}_1^l \cdots \overline{\omega}_M^l]^T$, $\underline{\omega}^r = [\underline{\omega}_1^r \cdots \underline{\omega}_M^r]^T$, and $\overline{\omega}^r = [\overline{\omega}_1^r \cdots \overline{\omega}_M^r]^T$ are used for clarity. Hence, the output of layer 4 is

$$O_j^{(4)} = \left[O_{jl}^{(4)} \ O_{jr}^{(4)}\right]^T = \left[\frac{\overline{\omega}_j^l \overline{O}_j^{(3)} + \underline{\omega}_j^l \underline{O}_j^{(3)}}{\overline{\omega}_j^l + \underline{\omega}_j^l} \quad \frac{\overline{\omega}_j^r \overline{O}_j^{(3)} + \underline{\omega}_j^r \underline{O}_j^{(3)}}{\overline{\omega}_j^r + \underline{\omega}_j^r}\right]^T \tag{6}$$

which calculates the left-most $O_{jl}^{(4)}$ and right-most $O_{jr}^{(4)}$. According to the type reduction is integrated in the adaptive network layer, KM algorithm is not necessary in the system [11].

Layer 5: TSK Layer

Due to the interval type-2 fuzzy sets are used for the antecedents and the interval type-1 fuzzy sets are used for the consequent sets of the type-2 TSK rules, C_{ji} are interval sets, that is , $C_{ji}=[c_{ji}-s_{ji}\ c_{ji}+s_{ji}]^T$, where c_{ji} denotes the center (mean) of C_{ji}, s_{ji} denotes the spread of C_{ji}, i=1, 2, …, n, and j=1, 2, …, M. Therefore, the consequent of *Rule j* is:

$$T_j = \left[T_j^l \ T_j^r\right]^T$$

$$= \left[(c_{j0} + \sum_{i=1}^{n} c_{ji}x_i) - (s_{j0} + \sum_{i=1}^{n} s_{ji}|x_i|) \quad (c_{j0} + \sum_{i=1}^{n} c_{ji}x_i) + (s_{j0} + \sum_{i=1}^{n} s_{ji}|x_i|)\right]^T . \tag{7}$$

Let

$$O_{TSK}^{(5)} = \left[T^l \ T^r\right]$$

$$= \int_{T_1 \in [T_1^l\ T_1^r]} \cdots \int_{T_M \in [T_M^l\ T_M^r]} \int_{O_1^{(4)} \in [O_{1l}^{(4)}\ O_{1r}^{(4)}]} \cdots \int_{O_M^{(4)} \in [O_{Ml}^{(4)}\ O_{Mr}^{(4)}]} 1 \Big/ \frac{\sum_{j=1}^{M} O_j^{(4)} T_j}{\sum_{j=1}^{M} O_j^{(4)}} , \tag{8}$$

and the output of layer 5 is

$$O^{(5)} = \left[O_l^{(5)} \ O_r^{(5)}\right]^T = \left[\frac{\sum_{j=1}^{M} O_{jl}^{(4)} T_j^l}{\sum_{j=1}^{M} O_{jl}^{(4)}} \quad \frac{\sum_{j=1}^{M} O_{jr}^{(4)} T_j^r}{\sum_{j=1}^{M} O_{jr}^{(4)}}\right]^T . \tag{9}$$

Layer 6: Output Layer

Layer 6 is the output layer which is used to implement the defuzzification operation. Each node is the actual output that will be pumped out this system. According to the TSK layer introduction, only $O_l^{(5)}$ and $O_r^{(5)}$ should be calculated. Therefore, the crisp output is

$$O^{(6)} = \frac{O_l^{(5)} + O_r^{(5)}}{2} . \tag{10}$$

III. Species-based Hybrid Algorithm Electromagnetism-like Mechanism Algorithm with Back-propagation Technique

This section introduces the proposed SEMBP for training aIT2FNS system. SEMBP combines the advantages of EM and BP algorithms with the notion of species and uniform method to have high speed convergence, less computation complexity, and global optimization. Figure 2 summarizes the hybrid learning algorithm SEMBP. There are four phases which are "Initialization," "Evaluation," "Species," and "IEM Operation" in SEMBP.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

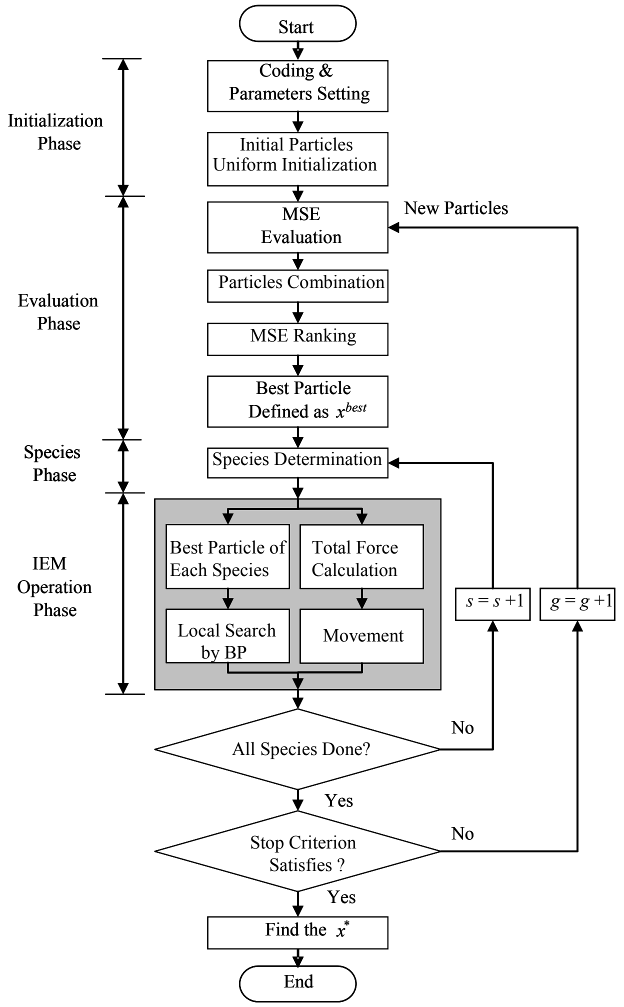


Figure 2: Description of SEMBP algorithm.

Initialization Phase

For SEMBP, each particle denotes a weighting vector

$$\mathbf{W} = \left[\overline{m}^l, \underline{m}^l, \overline{m}^r, \underline{m}^r, \overline{\sigma}^l, \underline{\sigma}^l, \overline{\sigma}^r, \underline{\sigma}^r, \gamma, \overline{\omega}, \underline{\omega}, c, s\right]^T \quad (11)$$

which decides the dimension of problem D . In this paper, we used the good lattice point method which is one of the uniform methods to construct the uniform arrays [12]. If the randomly initialization is used, the statistical analysis is necessary, i.e., repetition training to get average performance should be done. In this paper, we adopt the uniform method to avoid the repetition training. Thus, we can reduce the computation complexity. In addition, the uniform method has less probability to produce the outliers which may affect the results deeply and avoids the particles crowding in a region.

The good lattice point method of uniform method provides a series of uniform arrays for different n and q. $U_N(q^n)$ denotes the uniform array and transforms into the initial particles. There are N rows in $U_N(q^n)$ and each row represents a particle in R^n. In the Initialization Phase, based on $U_N(q^n)$, N particles can be generated as follows:

$$x_i = \left[x_{l1} + \frac{2u_{i1}-1}{2q}(x_{u1} - x_{l1}) \cdots x_{ln} + \frac{2u_{in}-1}{2q}(x_{un} - x_{ln})\right] \quad (12)$$

where x_i is the particles, x_{uk} and x_{lk} are the corresponding upper bound and lower bound, u_{ik} is the element of $U_N(q^n)$, i=1, 2, …, N, and k=1, 2, …, n.

Evaluation Phase

This phase is used to calculate the fitness values of entire particles and compare the fitness values. We retain the particles which have better MSE among the particles of generation g and g+1. This process can guarantee the better performance. Another task in this phase is to remove the redundant particles which have similar fitness and locate in Particles Combination. It can improve the efficiency of SEMBP by removing the redundant particles. Since the similar particles may converge to the similar location, we remain the best particle among the similar particles. As a matter of fact, the redundant particles do not contribute further to the improvement of convergence. The conditions of particles combination are

$$\mathrm{dis}(x_i, x_j) < 0.1 \times r_s \quad \text{and} \quad \left|\frac{\mathrm{MSE}(x_i) - \mathrm{MSE}(x_j)}{\mathrm{MSE}(x_i)}\right| < \mu_{th} \quad (13)$$

where dis(x_i, x_j) is the distance between x_i and x_j, r_s is the species radius, μ_{th} is the threshold and x_i is the particle, $i \neq j$.

Species Phase

The notion of species aims to identify multiple species via population and then determines a neighborhood best for each species. The dominating particle in each species is regarded as a neighborhood best called species seed. The species seed is always the fittest individual in the same species. All particles that fall within a distance from the species seed are classified as the same species. The definition of species depends on r_s, which denotes the radius that measured by distance from the center of a species to its boundary. Therefore, if r_s is small, many isolated species would be created in each generation. The small isolated particles species tend to prematurely converge to local minimum. If there are not sufficient numbers of particles in each species, the species will stop evolution. However, if r_s is large, it is possible to cover the entire variable range. In other words, the notion of species has no effect.

Subsequently, the particles in the population are sorted in decreasing order of fitness in MSE Ranking. As a matter of fact, the first particle in the ranking is the best-performing one and is denoted the species seed. The other particles in the population are checked in turn from best to worse and the particles whose distance between first species seed are smaller than r_s are categorized into the first species. If the particles do not fall within the radius of the first species seed, we select the particle which has minimum MSE to become a new species seed and the remaining particles are checked one by one that whether particles belong to the new species or not. Repeat the above steps until all particles are categorized. In this way, the species seeds and the number of species are generated in Species Phase.

Figure 3 provides an example to illustrate the working of this phase. In the example, the algorithm locates three species and the particles S_1, S_2, and S_3 are the species seeds. Note that there is overlap between first species and third species. Hence, the prior identified species which center is S_1 dominates the overlap which belongs to third species. In other words, the particle Q should belong to the species led by S_1.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

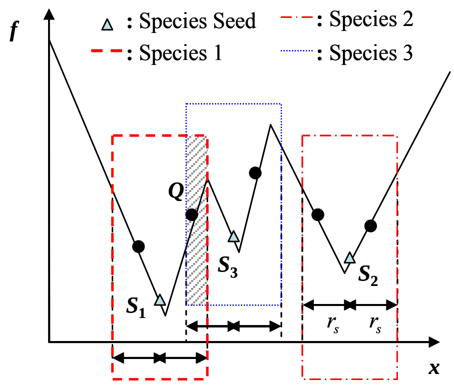


Figure 3: Example of how to determine the species in one-dimensioned problem.

IEM Operation Phase

There are three steps in the IEM operation phase: "Local Search of Best Particle by BP", "Total Force Calculation," and "Movement." In order to improve the random process, we choose the step length λ in Movement as one to accelerate the speed of convergence. After Species Determination, each sub-species proceeds Total Force Calculation and Movement which are the same as IEMBP's. Nevertheless, in contrast to the complete population, the subpopulations have less computation complexity in determining the electromagnetic charge of each particle. In complete population, we should compute $P_s\times(P_s-1)\times\ldots\times1$ times of charges in Total Force Calculation. But in the subpopulations, we only compute $\sum_S p_s\times(p_s-1)\times\cdots\times1$ times, where S is the number of species and p_s is the number of particles in subpopulations. And the best particle of sub-species processes local search by BP. In Local Search of Best Particle by BP, the gradient descent method is adopted to derive local search procedure of SEMBP for the aIT2FNS system. For clarification, we consider the single-output system and define the error cost function as

$$E(g)=\frac{1}{2}\sum_k e(k)^2 \tag{14}$$

where $e(k)=y_d(k)-\hat{y}(k)=y_d(k)-O^{(6)}(k)$, g is the index of generations, $\hat{y}(k)$ and $y_d(k)$ are the aIT2FNS's output and desired output for discrete time k, respectively. By using the gradient descent method, the parameters updated law is described as

$$\mathbf{W}(g+1)=\mathbf{W}(g)+\Delta\mathbf{W}(g)=\mathbf{W}(g)+\eta\left(-\frac{\partial E(g)}{\partial \mathbf{W}}\right) \tag{15}$$

where η is the learning rate. $\mathbf{W}=\left[\underline{\mathbf{W}},\overline{\mathbf{W}},\gamma,\mathbf{W}_\omega,C\right]^T$ are the adjustable parameters, where C is the parameters of TSK layer, $\mathbf{W}_\omega$ is the consequent weights, $\underline{\mathbf{W}}$ is the parameters of lower MFs, $\overline{\mathbf{W}}$ is upper MFs parameters and γ is the column vectors, i.e.,

$$C=\begin{bmatrix}c & s\end{bmatrix}^T, \tag{16}$$

$$\mathbf{W}_\omega=\begin{bmatrix}\underline{\omega}^l & \underline{\omega}^r & \overline{\omega}^l & \overline{\omega}^r\end{bmatrix}^T, \tag{17}$$

$$\underline{\mathbf{W}}=\begin{bmatrix}\underline{m}^l & \underline{m}^r & \underline{\sigma}^l & \underline{\sigma}^r\end{bmatrix}^T, \tag{18}$$

$$\overline{\mathbf{W}}=\begin{bmatrix}\overline{m}^l & \overline{m}^r & \overline{\sigma}^l & \overline{\sigma}^r\end{bmatrix}^T. \tag{19}$$

The update laws are similar to the results of [3]. For details, please refer to literature [3].

IV. Simulation Results

We apply the aIT2FNS with SEMBP for chaotic system identification. Consider the chaotic system describes in [1]

$$y_d(k)=-P\cdot y_d^2(k-1)+Q\cdot y_d(k-2)+1.0 \tag{20}$$

where P=1.4 and Q=0.3. For training the aIT2FNS system, we use the series-parallel learning scheme here. Herein, the inputs of the aIT2FNS are two input nodes for feeding the appropriate past values of $y_d(k-1)$ and $y_d(k-2)$, and the output of the aIT2FNS $\hat{y}(k)$ is the predicted result of system. The series-parallel training scheme is adopted as shown in Fig. 4. The approximated error is defined as follows

$$e(k)\equiv y_d(k)-\hat{y}(k). \tag{21}$$

The following mean-square-error (MSE) is adopted to be the performance index

$$MSE\equiv\frac{1}{T}\sum_{k=1}^{T}e^2(k) \tag{22}$$

where T is the number of training pattern.

In this simulation, we use the good lattice point method to construct the uniform array. We use the initial array $U_{61}(61^{60})$ to generate the initial parameters of aIT2FNS between [-1.5 1.5]. The learning rate of BP is selected to 0.1 and the threshold μ_{th} is selected to be 0.1. The initial condition is $[y_d(1), y_d(0)]^T$=[0.4, 0.4]^T. Parameters of SEMBP algorithm and aIT2FNS are chosen as in the following.

- Rule number (R): 2
- Network structure (layer 1~ layer 6): (2-4-2-4-2-1)
- Parameters number of aIT2FNS (D): 60
- Population size (P_s): 61
- Maximum generation (G): 20

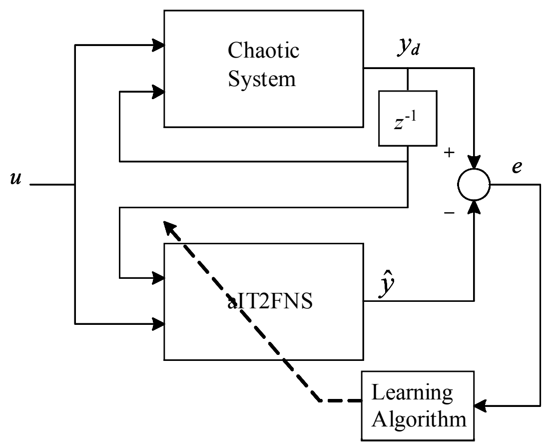


Figure 4: Series-parallel identification scheme.

The simulation results are described in Fig. 5 and Fig. 6. Figure 5(a) shows the phase plane of this chaotic system, whereas Figure 5(b) shows the identification result of aIT2FNS system. It can be observed that the SEMBP algorithm adjusts the parameters of aIT2FNS to predict system output accurately, and the aIT2FNS is similar to chaotic system. After training (20 generations), the MSE of aIT2FNS is 4.729×10^{-6}, which is better than the best results of other algorithms (as shown in Fig. 6). The MSE of BP which has fast convergence is 4.435×10^{-5}, but it achieves the

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

local minima. Obviously, the aIT2FNS system with SEMBP has better performance of accuracy, convergence and global optimum than the other algorithms. The final MFs are shown in Fig. 7(a)-(b). The constructed fuzzy rules are

R^1 : IF x_1 is $\widetilde{F}_{11}$ and x_2 is $\widetilde{F}_{21}$ THEN

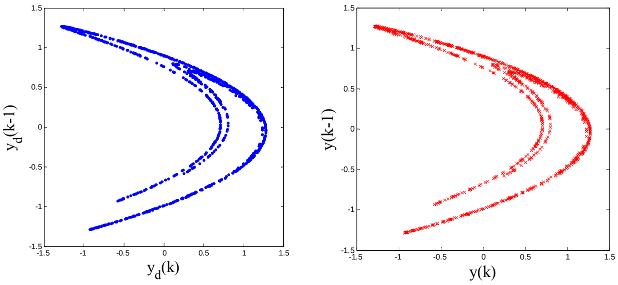
$Y_1 = [-1.465\ 1.463] + [0.299\ 1.702]x_1 + [-0.332\ 0.334]x_2$,

where $\overline{\omega} = [-0.256\ 1.235]$ and $\underline{\omega} = [-0.190\ 1.223]$.

R^2 : IF x_1 is $\widetilde{F}_{12}$ and x_2 is $\widetilde{F}_{22}$ THEN

$Y_2 = [0.236\ 0.947] + [-1.783\ 0.404]x_1 + [0.776\ 2.224]x_2$,

where $\overline{\omega} = [-0.987\ 0.454]$ and $\underline{\omega} = [-0.931\ 0.454]$.



(a) (b)

Figure 5: Phase plane plot of the example: (a) the chaotic system, (b) identification result of aIT2FNS.

Table 1 shows the comparison results of MSE in 20 training generations and the learning process is repeated for 20 runs. We can find that SEM takes 1665.1 second and EM takes 1306.1 second. It is obvious that SEM algorithm has better result than EM algorithm. Hence, we can know that using the species method has higher accuracy. In contrast with SEM and EM, SEMBP only takes 535.6 second which reduces the complexity of the calculation and promotes the simulated time efficiency. Although GA and PSO have similar computational complexity to SEMBP, but SEMBP has the better result. Referring to the Table 1, the MSE of SEMBP: 4.729×10^{-6} is smaller than those average MSEs of SEM: 1.381×10^{-3}; EM: 1.917×10^{-3}; PSO: 7.291×10^{-4}; GA: 8.420×10^{-4} and BP: 4.435×10^{-5}.

For the consideration of evaluations, Figure 8 shows the comparison results with different algorithms in the best MSE versus the evaluations. It can be seen that SEMBP does achieve better performance of MSE at the same evaluations. Thus, we can conclude that the SEMBP has the ability of high speed convergence, reduces the computational complexity and obtains global optimization.

The simulation results with different networks are shown in Table 2. In this simulation, we make the dimension D as large as possible under the constraint expecting that the larger rule number will result in better performance. If we choose 5 rule numbers (70 parameters) in IT2FNN, there are 10 peak values of upper MFs are not adjusted, and the remaining 60 parameters just satisfy the uniform array $U_{61}(61^{60})$. For the others which have less 60 parameters, we generate new uniform arrays by using different q. As we can see from Table 2, the network which has asymmetric type-2 MF is better than the others under the same rules. Due to the diversity of TSK, the aIT2FNS has more chance to get optimal solution. Obviously, the aIT2FNS with the TSK fuzzy rule and asymmetric MF has better performance than the other networks.

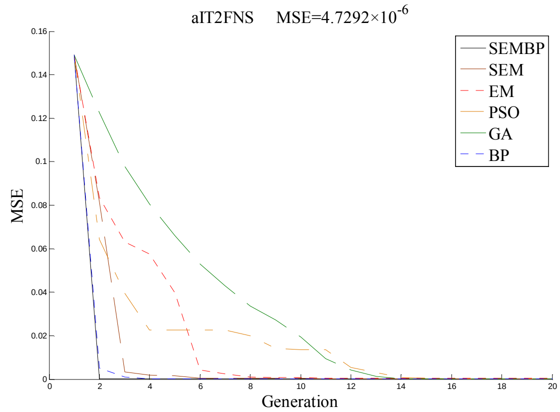
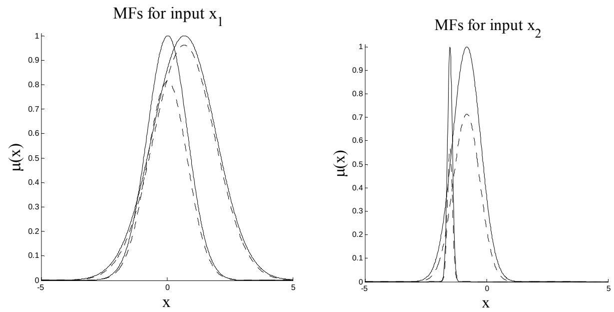


Figure 6: Simulation results: MSE in 20 generations. (−: SEMBP; —: SEM; --: EM; −-: PSO; —: GA; --: BP).



(a) (b)

Figure 7: The final membership functions: (a) MFs for x_1 after training, (b) MFs for x_2 after training.

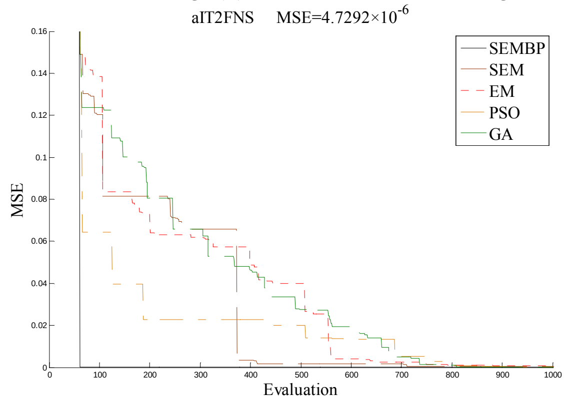


Figure 8: Comparison results in best MSE versus evaluations.

V. Conclusion

In this paper, we have proposed a novel hybrid learning algorithm SEMBP for training the aIT2FNS system. The corresponding SEMBP learning algorithm has been derived and used to train aIT2FNS for minimizing the training error. SEMBP combines EM with BP with the notion of species and uniform initialization to obtain higher convergent speed, less computational complexity, and global optimization. The aIT2FNS system, the consequent of the fuzzy rules is a linear combination of input variables, has the properties- fuzzy inference system, universal approximation and parameters convergence, *etc*. The asymmetric fuzzy MFs can improve the approximation accuracy and modeling capability. In addition, the uniform initialization is adopted to let the solution agents scatter over the feasible solution region evenly. Furthermore, the notion of species which can locate multiple optima provides bigger possibility of finding the global optimum. From the simulation results, we can observe that the proposed aIT2FNS with SEMBP algorithm has the ability of global optimization. And the simulation results also show that the aIT2FNS achieves better performance than the FNN, IT2FNN and IT2FNN-A systems. Performance comparisons with different categories of SEM, EM, PSO, GA and BP algorithms verify the effectiveness and efficiency of SEMBP. The example of chaotic system identification is proposed to show that SEMBP have the ability of global optimization and faster convergence.

REFERENCES

[1] C. H. Lee and C. C. Teng, "Identification and Control of Dynamic Systems Using Recurrent Fuzzy Neural Networks," *IEEE Trans. on Fuzzy Systems*, Vol. 8, No. 4, pp. 349-366, 2000.

[2] C. H. Lee and C. C. Teng, "Fine Tuning of Membership Functions for Fuzzy Neural Systems," *Asian Journal of Control*, Vol. 3, No. 3, pp. 216-225, 2001.

[3] C. H. Lee and H. Y. Pan, "Performance enhancement for neural fuzzy systems using asymmetric membership functions," *Fuzzy Sets and Systems,* Vol. 160, No. 7, pp. 949-971, 2009.

[4] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Upper Saddle River, Prentice-Hall, NJ, 2001.

[5] H. Hagras, "Type-2 FLCs: A New Generation of Fuzzy Controllers," *IEEE Computational Intelligence Magazine*, Vol. 2, No. 1, pp. 30-43, 2007.

[6] S. Horikawa, T. Furuhashi, and Y. Uchikawa, "On Fuzzy Modeling Using Fuzzy Neural Networks with the Back-propagation Algorithm," *IEEE Trans. on Neural Networks*, Vol. 3, No. 5, pp. 801-806, 1992.

[7] S. I. Birbil and S. C. Fang, "An Electromagnetism-like Mechanism for Global Optimization," *Journal of Global Optimization*, Vol. 25, No.3, pp. 263-282, 2003.

[8] M. Clerc and J. Kenney, "The Particle Swarm-explosion, Stability, and Convergence in A Multidimensional Complex Space," *IEEE Trans. on Evolutionary Computation*, Vol. 6, No.1, pp. 58-73, 2002.

[9] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, 1989.

[10] C. H. Lee, F. K. Chang, C. T. Kuo, and H. H. Chang, "A Hybrid of Electromagnetism-like Mechanism and Back-propagation Algorithms for Recurrent Neural Fuzzy Systems Design," *Int. J. of Systems Sciences*, (Revised), Jan. 2010.

[11] J. R. Castro, O. Castillo, P. Melin, and A. R. Díaz, "A Hybrid Learning Algorithm for A Class of Interval Type-2 Fuzzy Neural Networks," *Information Sciences*, Vol. 179, No. 3, pp. 2175-2193, 2009.

[12] K. T. Fang, *Number-theoretic Methods in Statistics*, Chapman & Hall, 1994.

Table 1: Comparison results of average performance in MSE and computational complexity with uniform initialization for by using different algorithms (G=20, D=60, P_s =61).

Algorithm	SEMBP	SEM	EM	PSO	GA	BP
Average Time	535.6	1665.1	1306.1	544.4	539.0	36.4
Average MSE	4.729×10^{-6}	1.381×10^{-3}	1.917×10^{-3}	7.291×10^{-4}	8.420×10^{-4}	4.435×10^{-5}
Best MSE		2.435×10^{-4}	4.442×10^{-4}	9.136×10^{-5}	1.816×10^{-5}	
Worst MSE		3.939×10^{-3}	3.056×10^{-3}	3.913×10^{-3}	4.002×10^{-3}	

Table 2: Comparison results in parameter number, computational complexity and MSE with uniform initialization and SEMBP by using different network and rule number (G=20).

Network	Network Structure	Rule No.	Parameter No.	Time	MSE
aIT2FNS	2-4-2-4-2-1	2	60	535.6	4.729×10^{-6}
IT2FNN-	2-4-2-1	2	44	573.1	1.604×10^{-4}
IT2FNN	2-4-2-1	2	28	481.0	1.652×10^{-2}
	2-10-5-1	5	70	760.2	1.870×10^{-4}
FNN	2-4-2-1	2	10	203.5	3.308×10^{-2}
	2-24-12-1	12	60	434.7	1.062×10^{-4}