Symbiotic Particle Swarm Optimization for Neural Fuzzy Controllers

Cheng-Hung Chen and Wen-Hsien Chen

Abstract—**This study proposes a symbiotic particle swarm optimization (SPSO) for the specific neural fuzzy controller (NFC). The specific NFC model using compensatory fuzzy operators of neural fuzzy networks makes fuzzy logic systems more adaptive and effective. The proposed SPSO adopts a multiple swarm scheme that uses each particle represents a single fuzzy rule and each particle in each swarm evolves separately to avoid falling in a local optimal solution. Furthermore, the SPSO embeds the symbiotic evolution scheme in a specific particle swarm optimization (PSO) to accelerate the search and increase global search capacity.**

Index Terms—**Water bath temperature system, neural fuzzy networks, symbiotic evolution, particle swarm optimization.**

I. INTRODUCTION

Neural fuzzy controllers (NFC) [1], [2] have been demonstrated to solving many engineering problems. They combine the capability of neural networks to learn from processes and the capability of fuzzy reasoning under linguistic information pertaining to numerical variables. On the other hand, recent development in genetic algorithms (GAs) has provided a method for neural fuzzy network design. Genetic fuzzy systems (GFSs) [3], [4] hybridize the approximate reasoning of fuzzy systems with the learning capability of genetic algorithms. Furthermore, a new optimization algorithm, called particle swarm optimization (PSO), appears to be better than the genetic algorithm. It is an evolutionary computation technique that was developed by Kennedy and Eberhart in 1995 [5]. The underlying motivation for the development of PSO algorithm is the social behavior of animals, such as bird flocking, fish schooling and swarm theory. PSO has been successfully applied to many optimization problems, such as control problems [6], [7].

This study proposes a symbiotic particle swarm optimization (SPSO) for the specific neural fuzzy controller (NFC). The specific NFC is based on our previous research [8] with adaptive compensatory fuzzy reasoning to dynamically adjust fuzzy operators. The proposed SPSO embeds the symbiotic evolution scheme in a particle swarm optimization (PSO) to accelerate the search and increase global search capacity. Unlike the GAs in a population as a full solution to a problem, symbiotic evolution [9] assumes that each individual in a population represents only a partial solution to a problem. Complex solutions combine several individuals in the population. The SPSO in which each particle represents a single fuzzy rule differs from original symbiotic evolution [8] to adopt a multiple swarm scheme. A fuzzy system with R-rules is constructed by selecting and combining R particles from each swarm, and allowing the rule itself to evolve.

II. STRUCTURE OF NEURAL FUZZY CONTROLLER

In this section, the structure of the specific NFC is introduced. Compensatory operators in the specific NFC model are used to optimize fuzzy logic reasoning and to select optimal fuzzy operators. Therefore, an effective NFC should not only adaptively adjust its fuzzy membership functions, it should also dynamically optimize adaptive fuzzy operators.
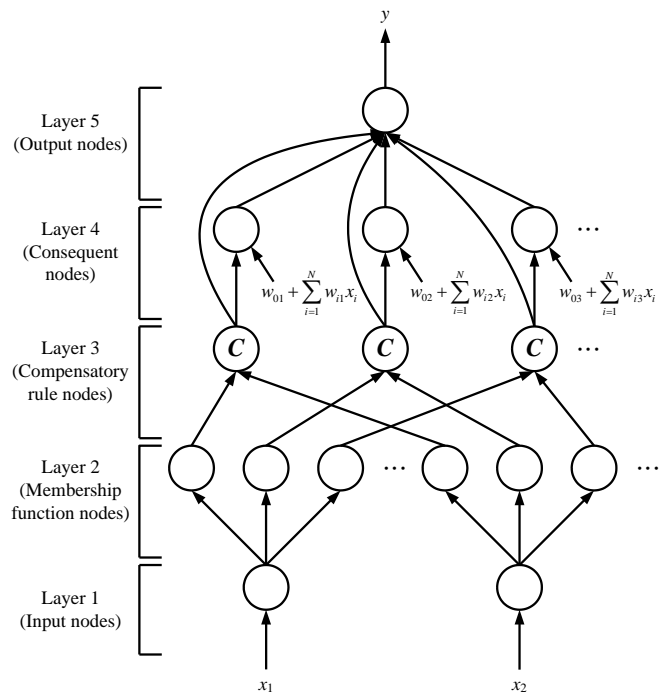

Fig. 1. Structure of the specific NFC model.

Fig. 1 shows the structure of the specific NFC model. The NFC realizes a fuzzy rule in the following form:

$$Rule-j : \text{IF}[x_1 \text{ is } A_{1j} \text{ and } x_2 \text{ is } A_{2j} \ldots \text{ and } x_i \text{ is } A_{ij} \text{ and } x_N \text{ is } A_{Nj}]^{1-\gamma_j+\frac{\gamma_j}{N}}$$
$$\text{THEN } y' \text{ is } w_{0j} + w_{1j}x_1 + w_{2j}x_2 + \ldots + w_{ij}x_i + w_{Nj}x_N \quad (1)$$

where x_i is the input variable; y' is the input variable; A_{ij} is the linguistic term of the precondition part; $\gamma_j \in [0,1]$ is the compensatory degree, w_{0j} and w_{ij} are the corresponding weight of feedback in the consequent part.

III. A SYMBIOTIC PARTICLE SWARM OPTIMIZATION FOR THE NFC MODEL

This section describes the proposed SPSO for the specific NFC model. The SPSO method comprises of three major components – create initial swarms, evaluate fitness value and update each particle. First, the initial swarms are created randomly before the evolution process begins. Second, the fitness value is evaluated for each particle by symbiotic evolution strategy to allow the rule itself to evolve in each swarm. Third, the each particle is updated using local best, global best and cooperative best in SPSO.

A. Create Initial Swarms

1) Coding step

The foremost step in SPSO is the coding of a fuzzy rule into a particle. Fig. 2 shows an example of the coding of parameters of a fuzzy rule into a particle where i and j represent the ith input variable and the jth rule, respectively. In this study, a Gaussian membership function is adopted with variables that represent the mean and deviation of the membership function. Fig. 2 represents a fuzzy rule given by Eq. (1), where m_{ij} and σ_{ij} are the mean and variance of a Gaussian membership function, respectively; γ_j is the compensatory factor, and w_{ij} is the corresponding parameter of the consequent part associated with the jth rule node. In this study, a real number represents the position of each particle.

Particle

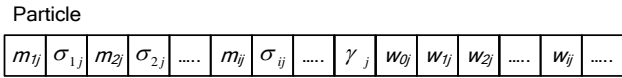| m_{1j} | σ_{1j} | m_{2j} | σ_{2j} | ..... | m_{ij} | σ_{ij} | ..... | γ_j | w_{0j} | w_{1j} | w_{2j} | ..... | w_{ij} | ..... |

Fig. 2. Coding a fuzzy rule into a particle.

2) Initial swarms

Before the SPSO method is applied, every position $x_{j,k}$ must be created randomly in the range [0, 1] in each swarm, where j=1, 2, …, R represents the jth swarm and k=1, 2, …, ps represents the kth particle.

B. Evaluate Fitness Value

This subsection presents a novel method of symbiotic evolution. As described above, in the symbiotic evolution, the fitness value of a rule (a particle) is computed as the sum of the fitness values of all the feasible combinations of that rule with all other randomly selected rules, and then dividing this sum by the total number of combinations. Fig. 3 shows the structure of the particle in the symbiotic evolution. In this figure, the best parameters of fuzzy system are reserved (i.e., the best particle) by the cooperative best (Cbest). The stepwise assignment of the fitness value is as follows.

Step 0: Divide the rules into swarms of size *ps*.

Step 1: Randomly select *R* fuzzy rules (particles) from each of the above swarms, and compose the fuzzy system using these *R* rules.

Step 2: Calculate fitness value of the composed fuzzy system. In this study, the fitness value is given by the follow formula;

$$Fitness\ value = \frac{1}{1+\sqrt{\frac{1}{D}\sum_{d=1}^{D}\left(y_d - \bar{y}_d\right)^2}} \quad (2)$$

where y_d represents the dth model output; $\bar{y}_d$ represents the desired output, and D represents the number of input data.

Step 3: Divide the fitness value by R and accumulate the divided fitness value to the fitness record of the R selected rules with their recorded fitness values initially set to zero.

Step 4: Repeat the above steps until each rule (particle) in each swarm has been selected a sufficient number of times, and record the number of fuzzy systems to which each particle has contributed.

Step 5: Divide the accumulated fitness of each particle by the number of times it has been selected.
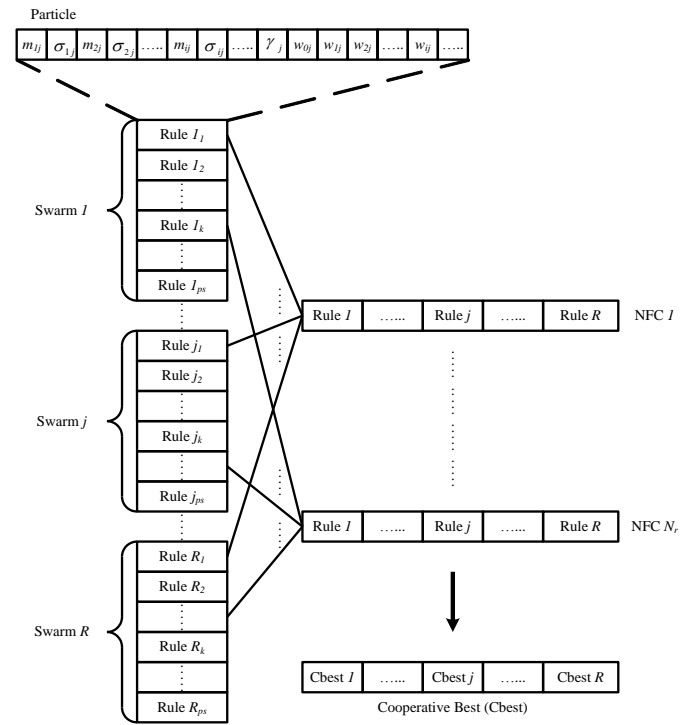

Fig. 3. Structure of particle in the proposed SPSO method.

C. Update Each Particle

Step 1: Update local best $L_{j,k}$, global best G_j, and cooperative best C_j

The local best position $L_{j,k}$ is the best previous position that yielded the best fitness value of the jth swarm of the kth particle, the global best position G_j is generated by the whole local best position and the cooperative best position C_j is created by parameters of the best composed fuzzy system. In step 1, the first step updates the local best position. Compare the fitness value of each current particle with that of its local best position. If the fitness value of the current particle exceeds those of its local best position, then the local best position is replaced with the position of the current particle. The second step updates the global best position. Compare the fitness value of all particles in their local best positions with that of the particle in the global best position. If fitness value of the particle in the local best position is better than those of the particles in the global best position, then the global best position is replaced with the current local best position.

$$L_{j,k} = \begin{cases} x_{j,k}, & if\ F\,(x_{j,k}) < F\,(L_{j,k}) \\ L_{j,k}, & if\ F\,(x_{j,k}) \geq F\,(L_{j,k}) \end{cases}$$

$$G_j = \arg\max_{L_{j,k}} F(L_{j,k}), \qquad 1 \leq k \leq ps \quad (3)$$

The third step updates the cooperative best position. Compare the fitness values of all composed fuzzy systems and the best fuzzy system. If the fitness value of one of all composed fuzzy systems exceeds those of the best fuzzy system, then the best fuzzy system is replaced with the composed fuzzy system in which the corresponding rule is the cooperative best (Cbest).

Step 2: Generate new swarms using $L_{j,k}$, G_j and C_j

The step updates velocity and position of each particle to generate the new swarms using Eqs. (4) and (5).

$$v_{j,k}(t+1) = \omega \cdot v_{j,k}(t) + \phi_1 \cdot Rand() \cdot (L_{j,k} - x_{j,k}) + \phi_2 \cdot Rand() \cdot (G_j - x_{j,k})$$
$$+ \phi_3 \cdot Rand() \cdot (C_j - x_{j,k})$$

$$\tag{4}$$

$$x_{j,k}(t+1) = x_{j,k}(t) + v_{j,k}(t+1) \tag{5}$$

where ω is the coefficient of inertia, ϕ_1 is the cognitive study, ϕ_2 is the society study, ϕ_3 is the group study, and $Rand()$ is generated from a uniform distribution in the range [0, 1].

IV. CONTROL OF WATER BATH TEMPERATURE SYSTEM

The goal of this section is to elucidate the control of the temperature of a water bath system according to,

$$\frac{dy(t)}{dt} = \frac{u(t)}{C} + \frac{Y_0 - y(t)}{T_R C} \tag{6}$$

where $y(t)$ is the output temperature of the system in ℃; $u(t)$ is the heat flowing into the system; Y_0 is room temperature; C is the equivalent thermal capacity of the system, and T_R is the equivalent thermal resistance between the borders of the system and the surroundings.

T_R and C are assumed to be essentially constant, and the system in Eq. (6) is rewritten in discrete-time form to some reasonable approximation. The system

$$y(k+1) = e^{-\alpha Ts} y(k) + \frac{\frac{\delta}{\alpha}(1 - e^{-\alpha Ts})}{1 + e^{0.5y(k)-40}} u(k) + [1 - e^{-\alpha Ts}]y_0 \tag{7}$$

is obtained, where α and δ are some constant values of TR and C. The system parameters used in this example are $\alpha = 1.0015e^{-4}$, $\delta = 8.67973e^{-3}$ and $Y_0 = 25.0$ ℃, which were obtained from a real water bath plant considered elsewhere [10]. The input $u(k)$ is limited to 0, and 5V represent voltage unit. The sampling period is $Ts=30$.

A sequence of random input signals $u_{rd}(k)$ limited to 0 and 5V is injected directly into the simulated system described in Eq.(7), using the online training scheme for the NFC-SPSO controller. The 120 training patterns are selected based on the input-outputs characteristics to cover the entire reference output. The temperature of the water is initially 25℃, and rises progressively when random input signals are injected.

In this simulation, the initial parameters before learning are given in Table I and five fuzzy rules are set to construct the NFC model. This dissertation compares the NFC-SPSO controller to the NFC-CPSO controller [11], the NFC-LPSO controller [12] and the NFC-PSO controller [5]. In the PSO,

LPSO and CPSO, the cognitive coefficient ϕ_1 was set to 2, the society coefficient ϕ_2 was set to 2, and the population size was set to 200. The coefficient ω of PSO was set to 0.4, the maximal and minimal weights of LPSO are set to 0.9 and 0.4 [12], respectively, and the constriction factor of CPSO is set to 0.729 [11]. Each of these controllers is applied to the water bath temperature control system. The performance measures include the set-points regulation, the influence of impulse noise, and a large parameter variation in the system, and the tracking capability of the controllers. Fig. 4 plots the learning curves of the best performance of the NFC-SPSO controller, the NFC-CPSO controller, the NFC-LPSO controller and the NFC-PSO controller for the fitness value, after the learning process of 10000 generations.

TABLE I: INITIAL PARAMETERS BEFORE LEARNING

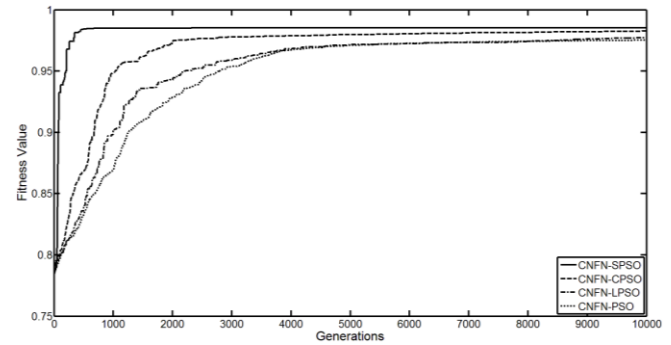| Parameter	Value
Population Size	50
Maximum Number of Generation	10000
ϕ_1, ϕ_2	1
ϕ_3	2
ω	0.4
Coding Type	Real Number

Fig. 4. Learning curves of best performance of the NFC-SPSO controller, the NFC-CPSO controller, the NFC-LPSO controller and the NFC-PSO controller.

In the simulations, the tracking capability of the NFC-SPSO controller with respect to ramp-reference signals is studied. Define

$$y_{ref}(k) = \begin{cases} 34\,℃ & \text{for} \quad k \le 30 \\ (34 + 0.5*(k-30))\,℃ & \text{for} \quad 30 < k \le 50 \\ (44 + 0.8*(k-50))\,℃ & \text{for} \quad 50 < k \le 70 \\ (60 + 0.5*(k-70))\,℃ & \text{for} \quad 70 < k \le 90 \\ 70\,℃ & \text{for} \quad 90 < k \le 120 \end{cases} \cdot \tag{8}$$

Fig. 5(a) presents the tracking performance of the NFC-SPSO controller. Fig. 5(b) presents the corresponding errors of the NFC-SPSO controller, the NFC-CPSO controller, the NFC-LPSO controller, and the NFC-PSO controller. To test their regulation performance, a performance index, the sum of absolute error (SAE), is defined by

$$SAE = \sum_k \left| y_{ref}(k) - y(k) \right| \tag{9}$$

where $y_{ref}(k)$ and $y(k)$ are the reference output and the actual output of the simulated system, respectively. The SAE

values of the NFC-SPSO controller, the NFC-CPSO controller, the NFC-LPSO controller, and the NFC-PSO controller are 45.18, 54.38, 68.41, and 76.33. The proposed NFC-SPSO controller has a much better SAE value of the tracking capability than the other controllers.
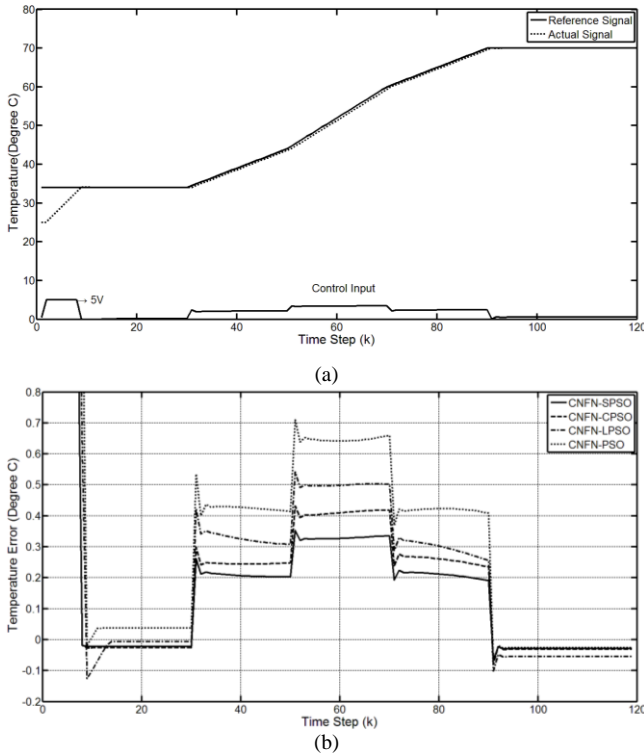

(a)

(b)

Fig. 5. (a) Tracking of NFC-SPSO controller when a change occurs in the water bath system. (b) Error curves of NFC-SPSO controller, NFC-CPSO controller, the NFC-LPSO controller, and NFC-PSO controller.

V. CONCLUSIONS

This study proposed a symbiotic particle swarm optimization (SPSO) for the specific NFC. The major novelty of the proposed SPSO learning algorithm uses a multiple swarm scheme to allow that each individual in each swarm evolves separately using a specific particle swarm optimization for constructing the NFC-SPSO method. Simulations demonstrate that the proposed NFC-SPSO method can obtain a better performance than other existing methods under some circumstances and has good generalization capability and robustness.

REFERENCES

[1] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent System*, NJ: Prentice-Hall, 1993.
[2] D. Nauck, F. Klawoon, and R. Kruse, *Foundations of Neuro-Fuzzy Systems*, New York: John Wiley, 1997.
[3] O. Cordon, F. Herrera, F. Hoffmann, and L. Magdalena, *Genetic Fuzzy Systems-Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, Singapore: World Scientific, 2001.
[4] P. P. Angelov, *Evolving Rule-Based Models: A Tool for Design of Flexible Adaptive Systems*, Physica-Verlag, Heidelberg, 2002.
[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. on Neural Networks*, 1995, pp. 1942-1948.
[6] Z. L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE Trans. on Energy Conversion*, vol. 19, no. 2, pp. 384-391, June 2004.
[7] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, and R. G. Harley, "Particle swarm optimization: basic concepts, variants and applications in power systems," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 171-195, Apr. 2008.
[8] C. H. Chen, "Efficient self-adaptive learning algorithm for TSK-type compensatory neural fuzzy networks," *International Journal of Fuzzy Systems*, vol. 14, no. 4, pp. 510-518, Dec. 2012.
[9] C. F. Juang, J. Y. Lin, and C. T. Lin, "Genetic reinforcement learning through symbiotic evolution for fuzzy controller design," *IEEE Trans. Syst., Man, Cybern. B*, vol. 30, no. 2, pp. 290-302, Apr. 2000.
[10] J. Tanomaru and S. Omatu, "Process control by on-line trained neural controllers," *IEEE Trans. on Ind. Electron.*, vol. 39, no. 6, pp. 511-521, Dec. 1992.
[11] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58-73, Feb. 2002.
[12] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE World Congr. Comput. Intell.*, 1998, pp. 69-73.

Cheng-Hung Chen received the B.S. and M.S. degrees in computer science and information engineering from the Chaoyang University of Technology, Taiwan, R.O.C., in 2002 and 2004, respectively, and the Ph.D. degree in electrical and control engineering from the National Chiao-Tung University, Taiwan, R.O.C., in 2008. Currently, he is an associate professor of Electrical Engineering Department, National Formosa University, Yunlin County, Taiwan, R.O.C. His current research interests are fuzzy systems, neural networks, evolutionary algorithms, intelligent control, pattern recognition, and image processing. He has authored or coauthored more than 50 papers published in the referred journals and conference proceedings. He is also a member of the Chinese Fuzzy Systems Association (CFSA), the Taiwanese Association for Artificial Intelligence (TAAI), and the IEEE Computational Intelligence Society.

Wen-Hsien Chen was born in Tainan, Taiwan, R.O.C. in 1990. He received the B.S. degree in electrical engineering from the Kun Shan University of Technology, Taiwan, R.O.C., in 2012. He is currently pursuing the M.S. degree in electrical engineering from the National Formosa University of Technology, Taiwan, R.O.C. His current research interests are fuzzy logic controller, intelligent control, and evolutionary algorithms.