

# Comparison of Two Interval Models for Fuzzy-Valued Genetic Algorithm

Hidehiko Okada

**Abstract** Genetic algorithm (GA) employs real numbers (or bit strings) as genotype values for solving real-valued optimization problems. The author previously proposed an extension of GA. The proposed method extends the processes of GA to handle fuzzy numbers as genotype values so that GA can be applied to fuzzy-valued optimization problems. The author has applied the FGA to the evolution of fuzzy-valued neural networks (FNN) and showed that FGA could evolve FNNs, which model fuzzy functions well, despite that the training (evolution) of the FNNs was not supervised. In the previous paper, fuzzy numbers as the genotype values were symmetric triangular ones. Each symmetric triangular fuzzy number can be specified by its lower and upper limit values or its center and width values, and thus the FGA can employ either of two models, the lower and upper (LU) model or the center and width (CW) model for specifying genotype values. Ability of the FGA in searching solutions may depend on the model, because the crossover and the mutation operations for the fuzzy genotypes with the LU model are slightly different from those operations with the CW model. In this paper, the author compares the two models to investigate which model contributes better for the FGA to find better solutions. Application of the FGA is evolutionary training of the FNNs. An experimental result shows that the CW model contributed slightly better than the LU model in evolving FNNs which model fuzzy functions.

**Keywords** Evolutionary algorithm · Genetic algorithm · Neural network · Neuroevolution · Fuzzy number

---

H. Okada (✉)

Faculty of Computer Science and Engineering, Kyoto Sangyo University, Kyoto, Japan  
e-mail: [hidehiko@cc.kyoto-su.ac.jp](mailto:hidehiko@cc.kyoto-su.ac.jp)

## 1 Introduction

Genetic algorithm (GA) [1], which is an instance of evolutionary algorithms [2], employs real numbers (or bit strings) as genotype values for solving real-valued optimization problems. The author previously proposed an extension of GA. The proposed method [3] extends the processes of GA to handle fuzzy numbers as genotype values so that GA can be directly applied to fuzzy-valued optimization problems. The author has applied the extended fuzzy-valued GA (FGA) to the evolution of fuzzy-valued neural networks (FNNs [4]) and showed that the FGA could evolve FNNs which model fuzzy functions well despite that the training (evolution) of the FNNs was not supervised [3].

In our previous paper, fuzzy numbers as the genotype values were symmetric triangular ones. Each symmetric triangular fuzzy number can be specified by its lower and upper limit values or its center and width values, and thus the FGA can employ either of two models, the lower and upper (LU) model or the center and width (CW) model for specifying genotype values. Ability of the FGA in searching solutions may depend on the model, because the crossover and the mutation operations for the fuzzy genotypes with the LU model are slightly different from those operations with the CW model. In this paper, the author compares the two models to investigate which model contributes better for the FGA to find better solutions. Application of the FGA is evolutionary training (neuroevolution [5]) of the FNNs.

## 2 Neural Networks with Fuzzy Weights and Biases

The FNN employed in the research is the same as in the literature [4], which is a three-layered feed-forward NN with fuzzy weights and biases. Figure 1 shows its structure. An FNN receives an input real vector  $\mathbf{x}$  and calculates its output fuzzy value  $O$  (for simplicity, the output layer includes a single unit) as follows [4]:

Input layer:

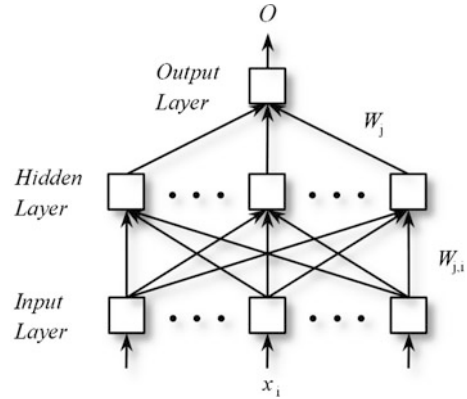
$$o_i = x_i. \quad (1)$$

Hidden layer:

$$\text{Net}_j = \sum_i W_{j,i} o_i + \theta_j, \quad (2)$$

$$O_j = f(\text{Net}_j). \quad (3)$$

**Fig. 1** Neural network with fuzzy weights and biases [4]



Output layer:

$$\text{Net} = \sum_j W_j o_j + \theta, \quad (4)$$

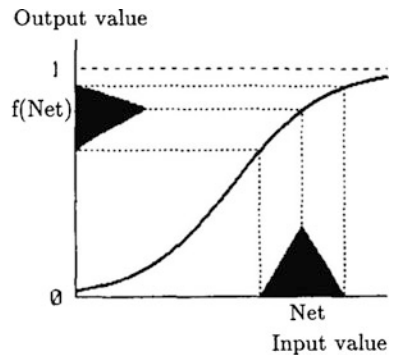
$$O = f(\text{Net}). \quad (5)$$

In Eqs. (1)–(5),  $x_i$  and  $o_i$  are real values, while  $\text{Net}_j$ ,  $\text{Net}$ ,  $W_{j,i}$ ,  $W_j$ ,  $\theta_j$ ,  $\theta$ ,  $O_j$  and  $O$  are fuzzy values.  $f(x)$  is the unit activation function which is typically the sigmoidal one:  $f(x) = 1/(1 + e^{-x})$ .  $f(x)$  maps a fuzzy input number to a fuzzy output number as illustrated in Fig. 2.

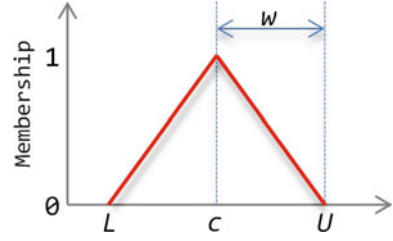
The feed-forward calculation of the FNN is based on the extension principle [6] and the interval arithmetic [7] (for more detail, see the literature [4]). Let us denote two closed intervals as  $A$  and  $B$ , where  $A = [a^L, a^U]$  and  $B = [b^L, b^U]$ . In this case,

$$\begin{aligned} A + B &= [a^L, a^U] + [b^L, b^U] \\ &= [a^L + b^L, a^U + b^U]. \end{aligned} \quad (6)$$

**Fig. 2** Input-output relation of each unit in the hidden and output layers [4]



**Fig. 3** Symmetric triangular fuzzy number and its real parameters



$$\begin{aligned} k \cdot A &= k \cdot [a^L, a^U] \\ &= [ka^L, ka^U], \text{ if } k \geq 0, \text{ else } [ka^U, ka^L]. \end{aligned} \quad (7)$$

$$\begin{aligned} A \cdot B &= [a^L, a^U] \cdot [b^L, b^U] \\ &= [\min(a^L b^L, a^L b^U, a^U b^L, a^U b^U), \max(a^L b^L, a^L b^U, a^U b^L, a^U b^U)]. \end{aligned} \quad (8)$$

The FNN includes  $mn + m$  weights (i.e.,  $mn$  weights between  $n$  input units and  $m$  hidden units, and  $m$  weights between  $m$  hidden units and an output unit) and  $m + 1$  biases (= the total number of units in the hidden and output layers). Thus, the FNN includes  $mn + 2m + 1$  fuzzy variables in total. the FGA handles these fuzzy variables as a genotype  $\mathbf{X} = (X_1, X_2, \dots, X_D)$  where  $X_i$  is a fuzzy number and  $D = mn + 2m + 1$ .

Suppose each  $X_i$  is a triangular fuzzy number (Fig. 3). In this case,  $X_i$  can be specified by its upper and lower limits or by its center and width:  $X_i = [x_i^L, x_i^U]$  or  $X_i = (x_i^c, x_i^w)$  where  $x_i^L, x_i^U, x_i^c, x_i^w$  denote the upper, lower, center and width of  $X_i$  respectively.

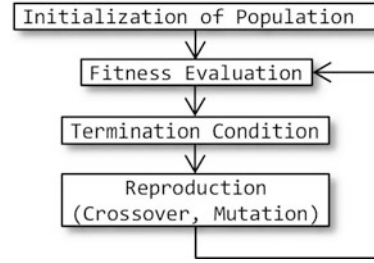
### 3 Genetic Algorithm with Fuzzy-Valued Genotypes

The FGA [3] includes the same processes as those in the ordinary GA (Fig. 4). Processes of initialization of population, fitness evaluation and reproduction are extended so that these processes can handle fuzzy-valued genotypes.

#### 3.1 Initialization of Population

In the initialization process,  $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_P$  are randomly initialized where  $P$  is the population size. Because the elements in  $\mathbf{X}_a$  (i.e.,  $X_{a,1}, X_{a,2}, \dots, X_{a,D}$ ) are weights and biases in a FNN in this research, smaller absolute values of  $X_{a,i}$  are preferable as initial values. Thus, the initial values for  $X_{a,i}$  are randomly sampled from the

**Fig. 4** Processes in the fuzzy-valued GA [3]



normal distribution  $N(0, \varepsilon)$  or uniformly from an interval  $[-\varepsilon, \varepsilon]$  where  $\varepsilon$  is a small positive number. In the case of employing the [lower, upper] model (the LU model), two values are sampled per  $X_{a,i} = [x_{a,i}^L, x_{a,i}^U]$ : the smaller (larger) one is set to  $x_{a,i}^L$  ( $x_{a,i}^U$ ). In the case of employing the (center, width) model (the CW model), two values are sampled per  $X_{a,i} = (x_{a,i}^c, x_{a,i}^w)$ : one of the two values is set to  $x_{a,i}^c$  and the absolute value of the other is set to  $x_{a,i}^w$ .

### 3.2 Fitness Evaluation

To evaluate fitness of an FNN as a phenotype instance of the corresponding genotype instance  $\mathbf{X}_a = (X_{a,1}, X_{a,2}, \dots, X_{a,D})$  where  $\mathbf{X}_a \in \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_P\}$ , the FNN is supplied with several samples of input real vectors and calculates output values. The input values are sampled within the variable domain of application problem. Fitness of the genotype instance  $\mathbf{X}_a$  is evaluated based on the output values. The method for scoring the fitness based on the output values depends on the problem to which the FNN is applied. For example, in a case where the FNN is applied to controlling an automated system, some performance measure of the system can be used as the fitness score of the genotype instance corresponding to the FNN.

### 3.3 Crossover

Let us denote genotypes of two parents as  $\mathbf{X}_a, \mathbf{X}_b$  and an offspring genotype as  $\mathbf{X}_z$ .  $\mathbf{X}_a$  and  $\mathbf{X}_b$  can be sampled from the population in the same manner as the ordinary GA. In the case of employing the LU model,

$$\mathbf{X}_a = (X_{a,1}, X_{a,2}, \dots, X_{a,D}), X_{a,i} = [x_{a,i}^L, x_{a,i}^U], \quad (9)$$

$$\mathbf{X}_b = (X_{b,1}, X_{b,2}, \dots, X_{b,D}), X_{b,i} = [x_{b,i}^L, x_{b,i}^U], \quad (10)$$

$$\mathbf{X}_z = (X_{z,1}, X_{z,2}, \dots, X_{z,D}), X_{z,i} = [x_{z,i}^L, x_{z,i}^U]. \quad (11)$$

Values of  $x_{z,i}^L$  and  $x_{z,i}^U$  in the offspring  $\mathbf{X}_z$  can be determined by applying the blend crossover [8] for the real GA:  $x_{z,i}^L$  is randomly sampled from the interval  $[\min(x_{a,i}^L, x_{b,i}^L) - \alpha|x_{a,i}^L - x_{b,i}^L|, \max(x_{a,i}^L, x_{b,i}^L) + \alpha|x_{a,i}^L - x_{b,i}^L|]$ , where  $\min(x, y)$  ( $\max(x, y)$ ) is the smaller (larger) of  $x$  and  $y$ . Similarly,  $x_{z,i}^U$  is randomly sampled from the interval  $[\min(x_{a,i}^U, x_{b,i}^U) - \alpha|x_{a,i}^U - x_{b,i}^U|, \max(x_{a,i}^U, x_{b,i}^U) + \alpha|x_{a,i}^U - x_{b,i}^U|]$ . Note that  $x_{z,i}^U$  must not be smaller than  $x_{z,i}^L$  because  $x_{z,i}^L$  and  $x_{z,i}^U$  are the lower and upper limits of the fuzzy number  $X_{z,i}$ . If  $x_{z,i}^U$  becomes smaller than  $x_{z,i}^L$  as the result of applying the blend crossover operations, then  $x_{z,i}^L$  and  $x_{z,i}^U$  must be repaired so that  $X_{z,i}$  is valid. The repair method can be either of the followings:

- the value of  $x_{z,i}^U$  is assigned to  $x_{z,i}^L$ ,
- the value of  $x_{z,i}^L$  is assigned to  $x_{z,i}^U$ ,
- the mean value of  $x_{z,i}^L$  and  $x_{z,i}^U$  is calculated and assigned to both of  $x_{z,i}^L$  and  $x_{z,i}^U$ , or
- the two values for  $x_{z,i}^L$  and  $x_{z,i}^U$  are switched.

In the case of employing the CW model,

$$\mathbf{X}_a = (X_{a,1}, X_{a,2}, \dots, X_{a,D}), X_{a,i} = (x_{a,i}^c, x_{a,i}^w), \quad (12)$$

$$\mathbf{X}_b = (X_{b,1}, X_{b,2}, \dots, X_{b,D}), X_{b,i} = (x_{b,i}^c, x_{b,i}^w), \quad (13)$$

$$\mathbf{X}_z = (X_{z,1}, X_{z,2}, \dots, X_{z,D}), X_{z,i} = (x_{z,i}^c, x_{z,i}^w). \quad (14)$$

Values of  $x_{z,i}^c$  and  $x_{z,i}^w$  in the offspring  $\mathbf{X}_z$  can be determined in the same manner as those for the LU model:  $x_{z,i}^c$  is randomly sampled from the interval  $[\min(x_{a,i}^c, x_{b,i}^c) - \alpha|x_{a,i}^c - x_{b,i}^c|, \max(x_{a,i}^c, x_{b,i}^c) + \alpha|x_{a,i}^c - x_{b,i}^c|]$ . Similarly,  $x_{z,i}^w$  is randomly sampled from the interval  $[\min(x_{a,i}^w, x_{b,i}^w) - \alpha|x_{a,i}^w - x_{b,i}^w|, \max(x_{a,i}^w, x_{b,i}^w) + \alpha|x_{a,i}^w - x_{b,i}^w|]$ . Note again that  $x_{z,i}^w$  must not be negative because  $x_{z,i}^w$  is the width of the fuzzy number  $X_{z,i}$ . If  $x_{z,i}^w$  becomes negative as the result of applying the blend crossover operation, then  $x_{z,i}^w$  must be repaired so that  $X_{z,i}$  is valid. The repair method can be either of the followings:

- the value of  $x_{z,i}^w$  is assigned to 0, or
- the absolute value of  $x_{z,i}^w$  is assigned to  $x_{z,i}^w$ .

### 3.4 Mutation

Values in the offspring genotypes are mutated under the specified mutation probability. In the FGA, each offspring  $X_z$  is a vector  $(X_{z,1}, X_{z,2}, \dots, X_{z,D})$  where  $X_{z,i}$  is a fuzzy number specified by the two real parameters:  $X_{z,i} = [x_{z,i}^L, x_{z,i}^U]$  or  $X_{z,i} = (x_{z,i}^c, x_{z,i}^w)$ . Each element of  $x_{z,i}^L, x_{z,i}^U$  (or  $x_{z,i}^c, x_{z,i}^w$ ) selected under the probability is mutated by being added (or replaced) with a random real number  $r$  to the current value, where  $r$  is randomly sampled from the normal distribution  $N(0, \delta)$  or uniformly from an interval  $[-\delta, \delta]$  and  $\delta$  is also a small positive number as  $\varepsilon$ . After the mutation of  $X_{z,i}$ ,  $x_{z,i}^U$  may become smaller than  $x_{z,i}^L$  (with the LU model) or  $x_{z,i}^w$  may become negative (with the CW model). Such invalid fuzzy numbers are repaired by the same method applied in the crossover process.

## 4 Comparison of LU/CW Models for Fuzzy Genotype Values in FGA

As described in Sect. 3.3, the constraints for the two real parameters of a symmetric triangular fuzzy number (i.e., the lower and upper values or the center and width values) are different, and thus the methods for repairing constraint-violating values are also different between the LU and CW models. This difference may affect the performance of FGA in searching solutions. To compare the performances between the two models, FGA with each of the two models is applied to the same problem. As the application problem, evolution of FNNs is employed. FGA is challenged to evolve FNNs which better model a target fuzzy function.

In the experiment, the author designs a function and employs it as the modeling targets for FNNs. For simplicity, the input  $x$  of the function is a real value (so that the FNN includes only a single input unit) and  $0 \leq x \leq 1$ , as in the literature [4]. The output values of the function are symmetric triangular fuzzy numbers. The function  $F(x) = [F(x)^L, F(x)^U]$  is as follows:

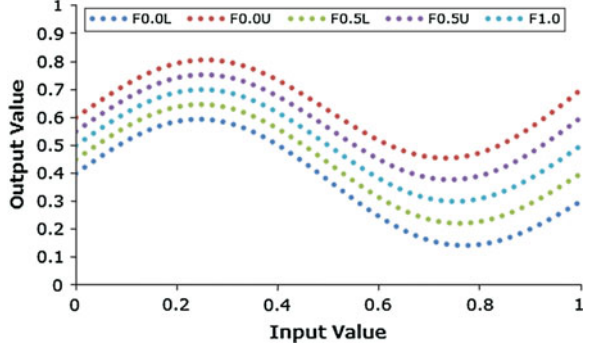
$$F(x)^L = 0.2 \sin(2\pi x) - 0.1x^2 + 0.4, \quad (15)$$

$$F(x)^U = 0.2 \sin(2\pi x) + 0.1x^2 + 0.6. \quad (16)$$

Figure 5 shows the shape of the target fuzzy function  $F(x)$ , where:

- F0.0L and F0.0U denote  $F(x)^L$  and  $F(x)^U$ , i.e., the lower and upper limits of the support interval of  $F(x)$ ,
- F0.5L and F0.5 denote the lower and upper limits of the 0.5-level interval of  $F(x)$ , i.e.,  $F(x)|_{0.5}$ , and
- F1.0 denotes the peak of  $F(x)$ , i.e.,  $F(x)|_{1.0}$ .

**Fig. 5** Target fuzzy function  $F(x)$



The FNN and the FGA are designed as follows.

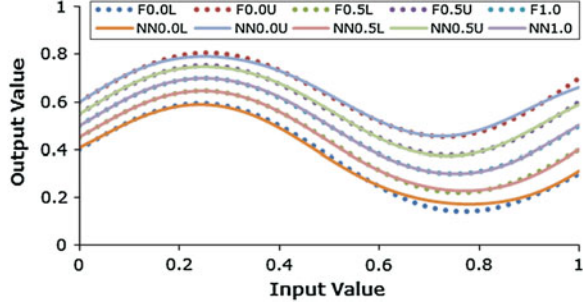
- FNN:
  - #units: 1 input, 10 hidden, 1 output.
  - Initial values for  $x_{j,i}^L$ ,  $x_{j,i}^U$ ,  $x_{j,i}^c$ : uniformly random within  $[-0.01, 0.01]$ .
  - Initial values for  $x_{j,i}^w$ : uniformly random within  $[0.0, 0.01]$ .
  - $-10.0 \leq x_{j,i}^L, x_{j,i}^U, x_{j,i}^c \leq 10.0$ .
  - $0.0 \leq x_{j,i}^w \leq 10.0$ .
- FGA:
  - #Total FNNs evolved: 1,000,000.
  - Population size and #generation: (100 and 10,000) or (500 and 2,000).
  - $\alpha$  for the blend crossover: 0.5.
  - Mutation probability: 0.01 for each of the elements  $X_{i,1}, X_{i,2}, \dots, X_{i,D}$  in a genotype instance  $X_i$ .
  - Random values for mutation:  $N(0, 1)$  for  $x_{j,i}^L$ ,  $x_{j,i}^U$ ,  $x_{j,i}^c$  and  $|N(0, 1)|$  for  $x_{j,i}^w$ .
  - Elitism: best 10 elite genotype instances are copied to the next generation.
  - Tournament size for sampling two parent genotype instances: 5 % of the population size.

The number of generations is 10,000 (or 2,000) for FGA with 100 (or 500) solutions so that the total number of FNNs evolved is consistently 1,000,000.

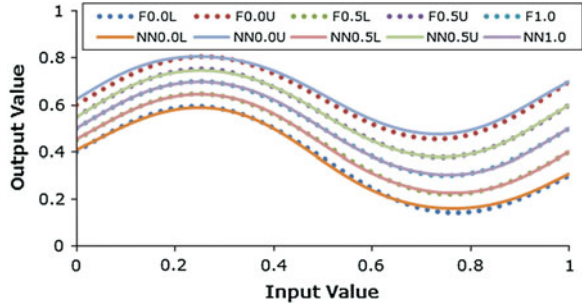
Genotype instances  $X_1, X_2, \dots, X_P$ , are ranked by utilizing the same cost function as that in literature [4]. As the values for the  $h$ -level intervals of fuzzy numbers, the author employs  $h \in \{0.2, 0.4, \dots, 1.0\}$  in this experiment. An FNN which corresponds to a genotype instance  $X_i$  is supplied with a real input value  $x_r$  and calculates its output fuzzy number  $O_r$ .  $x_r$  is sampled within the input domain  $[0, 1]$  as  $x_r = \{0.0, 0.01, 0.02, \dots, 1.0\}$ . Besides, each value of  $x_r$  is supplied to the target function  $F(x)$  and the output fuzzy number  $F(x_r)$  is obtained. Then, the error  $e_r$  for  $x_r$  is calculated as



**Fig. 6** Output fuzzy function of the FNN evolved by FGA with the LU model. The error score was 6.4E-3



**Fig. 7** Output fuzzy function of the FNN evolved by FGA with the CW model. The error score was 4.5E-3



$$e_r = \sum_h h((o_{r,h}^L - f_{r,h}^L)^2 + (o_{r,h}^U - f_{r,h}^U)^2) \quad (17)$$

where,

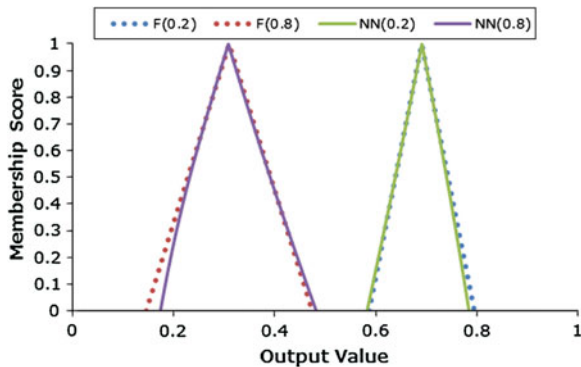
- $o_{r,h}^L$  and  $o_{r,h}^U$  are the lower and upper limits of the  $h$ -level interval of  $O_r$ , i.e.,  $O_r|_h = [o_{r,h}^L, o_{r,h}^U]$ , and
- $f_{r,h}^L$  and  $f_{r,h}^U$  are the lower and upper limits of the  $h$ -level interval of  $F(x_r)$ , i.e.,  $F(x_r)|_h = [f_{r,h}^L, f_{r,h}^U]$ .

For each genotype instance  $X_i$ ,  $e_r$  is calculated 101 times ( $e_0, e_1, \dots, e_{100}$ ) for the 101 input values  $x_r = \{0.0, 0.01, 0.02, \dots, 1.0\}$ , and the sum of  $e_r$  is used for ranking  $X_i$ . An instance with a smaller sum of  $e_r$  is ranked better. Note that  $e_r$  scores are not utilized for calculating the values of updating the weights and biases but only for ranking the genotype instances: the target function  $F(x)$  is completely hidden from the FGA processes.

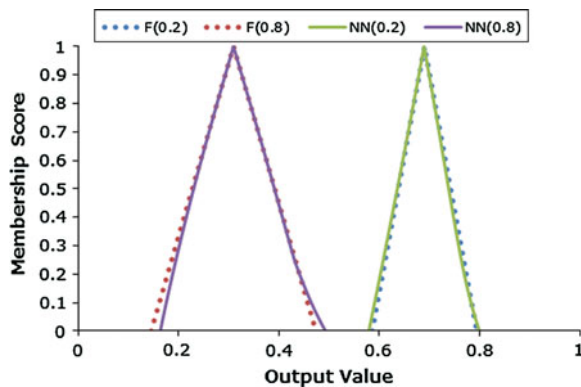
Figures 6, 7, 8, 9 and 10 show the results of this experiment. Figures 6 and 7 show the output fuzzy function of the best FNN among the total 10,000,000 FNNs ( $= [1,000,000 \text{ FNNs in each run}] * [5 \text{ runs}] * [\text{two values for population sizes}]$ ) evolved by the FGA with the LU model (Figure 6) and the CW model (Figure 7) respectively. In Figs. 6 and 7,

- F0.0L, F0.0U, F0.5L, F0.5U and F1.0 are the same as those in Fig. 5,

**Fig. 8** Output fuzzy numbers of the FNN evolved by FGA with the LU model and the target fuzzy numbers  $F(x)$  for the inputs values of 0.2 and 0.8



**Fig. 9** Output fuzzy numbers of the FNN evolved by FGA with the CW model and the target fuzzy numbers  $F(x)$  for the inputs values of 0.2 and 0.8

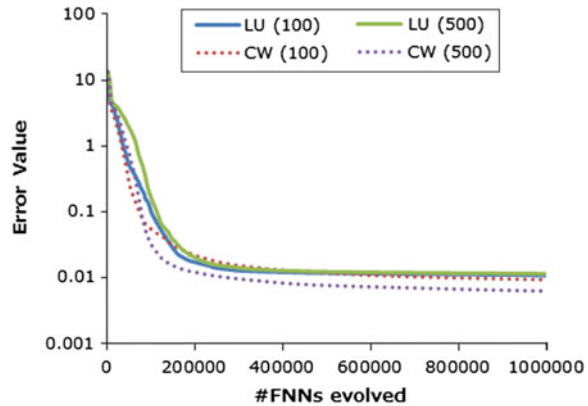


- NN0.0L and NN0.0U denote the lower and upper limits of the support interval of the FNN output fuzzy number,
- NN0.5L and NN0.5U denote the lower and upper limits of the 0.5-level interval of the FNN output fuzzy number, and
- NN1.0 denotes the peak of the FNN output fuzzy number.

Figures 8 and 9 show the membership functions of the FNN output fuzzy numbers and the target function  $F(x)$  for the input values  $x = 0.2$  and  $x = 0.8$ . Figures 8 and 9 show the result with the LU and CW models respectively. The shapes of the FNN output fuzzy numbers (the solid curves in Figs. 8 and 9) are very similar to those of the target fuzzy numbers (the dotted lines in Figs. 8 and 9). Figures 6, 7, 8 and 9 reveal that both of the best FNNs evolved with the LU and CW models fit to the hidden target function  $F(x)$  very well, despite the fact that no training data is explicitly provided.

Figure 10 shows the error value of the best FNN among each number of FNNs evolved (e.g., 500,000 FNNs are evolved in total at the 5,000th generation with the population size of 100). In Fig. 10, “LU (100)” denotes the result with the LU

**Fig. 10** Error value of the best FNN at each number of FNNs evolved



model and the population size of 100. “LU (500)”, “CW (100)” and “CW (500)” denote their results in the same manner as “LU (100)”. The error values are the averaged ones over the 5 runs. Figure 10 reveals that the CW model contributed slightly better to the FGA than the LU model did with the population size of both 100 and 500 (i.e., after the 1,000,000 FNNs evolved, the dotted curves for the CW model went below the solid curves for the LU model).

## 5 Conclusion

The two models for describing symmetric triangular fuzzy numbers, i.e., the LU model and the CW model, were compared so that which model will contribute better to the fuzzy-valued genetic algorithm. The experiment revealed that both model contributed well and the CW model was slightly better than the LU model.

In the future work, the author will further study the ability of the FGA by applying it to fuzzy optimization problems other than the evolution of fuzzy neural networks. Besides the author will further investigate the FGA where the genotype values are not symmetric triangular fuzzy numbers but Gaussian ones, asymmetric triangular ones, trapezoidal ones, etc.

**Acknowledgment** This research was supported by Kyoto Sangyo University Research Grant.

## References

1. Goldberg, D.E.: Genetic Algorithms in Search Optimization and Machine Learning. Kluwer Academic Publishers, Norwell (1989)
2. Back, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford Univ Press, Oxford (1996)

3. Okada, H., Matsuse, T., Wada, T.: GA with fuzzy-valued genotypes and its application to neuroevolution. In: Proceedings of Asia Pacific Symposium of Intelligent and Evolutionary Systems (IES) 2012, pp. 15–18 (2012)
4. Ishibuchi, H., Tanaka, H., Okada, H.: Fuzzy neural networks with fuzzy weights and fuzzy biases. In: Proceedings of IEEE International Conferences on Neural Networks, pp. 1650–1655 (1993)
5. Yao, X.: Evolving artificial neural networks. In: Proceedings of the IEEE, vol. 87, no. 9, pp. 1423–1447 (1999)
6. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning—I, II, and III. In: Information Science, vol. 8, pp. 199–249, pp. 301–357 and vol. 9, pp. 43–80 (1975)
7. Alefeld, G., Herzberger, J.: Introduction to Interval Computation Academic Press, New York (1983)
8. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. In: Whitley, D.L. (ed.) Foundation of Genetic Algorithms 2, pp. 187–202 (1993)

<http://www.springer.com/978-4-431-55104-1>

Recent Advances in Natural Computing

Selected Results from the IWNC 7 Symposium

Suzuki, Y.; Hagiya, M. (Eds.)

2015, VIII, 132 p. 39 illus., 18 illus. in color., Hardcover

ISBN: 978-4-431-55104-1