International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:7, No:10, 2013

Comparison of Two Interval Models for Interval-Valued Differential Evolution

Hidehiko Okada

***Abstract*—**The author previously proposed an extension of differential evolution. The proposed method extends the processes of DE to handle interval numbers as genotype values so that DE can be applied to interval-valued optimization problems. The interval DE can employ either of two interval models, the lower and upper model or the center and width model, for specifying genotype values. Ability of the interval DE in searching for solutions may depend on the model. In this paper, the author compares the two models to investigate which model contributes better for the interval DE to find better solutions. Application of the interval DE is evolutionary training of interval-valued neural networks. A result of preliminary study indicates that the CW model is better than the LU model: the interval DE with the CW model could evolve better neural networks.

***Keywords*—**Evolutionary algorithms, differential evolution, neural network, neuroevolution, interval arithmetic.

I. Introduction

DIFFERENTIAL EVOLUTION (DE) [1], which is an instance of evolutionary algorithms [2], employs real numbers as genotype values for solving real-valued optimization problems. The author previously proposed an extension of DE. The proposed method [3] extends the processes of DE to handle interval numbers as genotype values so that DE can be applied to interval-valued optimization problems. The author has applied the extended interval-valued DE (IDE) to the evolution of interval-valued neural networks (INN [4]) and showed that IDE could evolve INNs which model interval target functions well despite that no training data was explicitly provided [5].

An interval value can be specified by its lower and upper limit values or its center and width values, and thus the IDE can employ either of two interval models, the *lower and upper* model (LU) or the *center and width* model(CW) for specifying genotype values. Ability of the IDE in searching for solutions may depend on the model. In this paper, the author compares the two models to investigate which model contributes better for the IDE to find solutions. Application of the IDE is the same as that in our previous paper [5], i.e., evolutionary training of the INNs.

II. Neural Networks with Interval Weights and Biases

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

Hidehiko Okada is with Department of Intelligent Systems, Faculty of Computer Science and Engineering, Kyoto Sangyo University, Japan (e-mail: hidehiko@cc.kyoto-su.ac.jp).

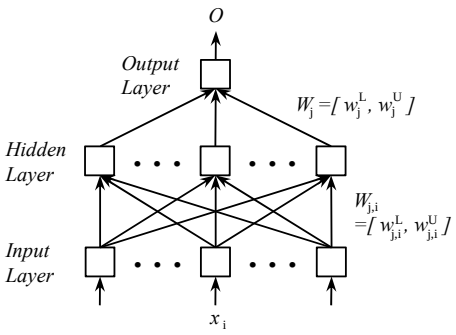

Fig. 1 Neural network with interval weights and biases [4]

Input layer:

$$o_i=x_i. \tag{1}$$

Hidden layer:

$$Net_j=\Sigma_i W_{j,i}o_i+\Theta_j, \tag{2}$$

$$O_j=f(Net_j). \tag{3}$$

Output layer:

$$Net=\Sigma_j W_j O_j+\Theta, \tag{4}$$

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

In (1)-(5), x_i and o_i are real values, while Net_j, Net, $W_{j,i}$, W_j, Θ_j, Θ, O_j and $O$are interval values. $f(x)$ is the unit activation function which is typically the sigmoid alone: $f(x)=1/(1+e^{-x})$. $f(x)$ maps an interval input to an interval output as illustrated in Fig. 2.

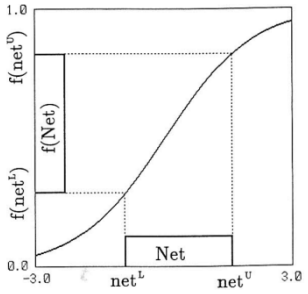

Fig. 2 Input-output relation of each unit in the hidden and output layers [4]

For the feed forward calculation of the INN, the interval arithmetic [6] is utilized. 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,

$$A + B = [a^L, a^U] + [b^L, b^U] = [a^L + b^L, a^U + b^U]. \quad (6)$$

$$k \cdot A = k \cdot [a^L, a^U] = [ka^L, ka^U], \text{ if } k \geq 0, \text{ or } [ka^U, ka^L], \text{ if } k < 0 \quad (7)$$

$$A \cdot B = [a^L, a^U] \cdot [b^L, b^U] = [\min(a^L b^L, a^L b^U, a^U b^L, a^U b^U), \max(a^L b^L, a^L b^U, a^U b^L, a^U b^U)]. \quad (8)$$

The INN includes $mn+m$ weights (i.e., nm 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 INN includes $mn+2m+1$ interval variables in total. Our IDE handles these interval variables as a genotype $\boldsymbol{X} = (X_1, X_2, \ldots, X_D)$ where X_i is an interval and $D = mn+2m+1$. Each X_i can be specified by its upper and lower real values 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 and x_i^w denote the upper, lower, center and width of X_i respectively.

III. Differential Evolution with Interval-Valued Genotypes

The IDE consists of the same processes as those in the ordinary DE. Processes of initialization of populations, reproduction and fitness evaluation are extended so that these processes can handle interval-valued genotypes.

A. Initialization of Population

In the initialization process, $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_P$ are randomly initialized where P is the population size. Because the elements in $\boldsymbol{X}_a$ (i.e., $X_{a,1}, X_{a,2}, \ldots, X_{a,D}$) are interval weights and biases in an INN 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 normal distribution $N(0, \varepsilon)$ or uniformly from an interval $[-\varepsilon, \varepsilon]$ where ε is a small positive number. In the case of the [lower, upper] model, two values are sampled per $X_{a,i}$: the smaller (larger) one is set to $x_{a,i}^L$ ($x_{a,i}^U$). In the case of the (center, width) model, two values are sampled per $X_{a,i}$: one of the two values is set to $x_{a,i}^c$ and the absolute value of the other is set to $x_{a,i}^w$ because $x_{a,i}^w$ (the width of the interval $X_{a,i}$) must be non-negative.

B. Fitness Evaluation

To evaluate fitness of an INN as a phenotype instance of the corresponding genotype instance $\boldsymbol{X}_a = (X_{a,1}, X_{a,2}, \ldots, X_{a,D})$ where $\boldsymbol{X}_a \in \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_P\}$, the INN is supplied with input real vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_S$ and calculates output interval values $O_1, O_2, \ldots, O_S$. $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_S$ are sampled within the variable domain of application. Fitness of the genotype instance $\boldsymbol{X}_a$ is evaluated based on $O_1, O_2, \ldots, O_S$. The method for calculating the fitness score depends on the task to which the INN is applied. For example, in a case where the INN is applied to controlling an automated robot system, some performance measure of the system can be used as the fitness score of the genotype instance corresponding to the INN.

C. Reproduction

There are several variants of reproduction methods for DE. This paper describes the proposed extension of DE/rand/1/bin. Suppose $\boldsymbol{X}_a$ is a parent, $\boldsymbol{X}_a'$ is an offspring from $\boldsymbol{X}_a$, $\boldsymbol{Y}_a$ is the donor vector for $\boldsymbol{X}_a$, $\boldsymbol{X}_{b1}$, $\boldsymbol{X}_{b2}$ and $\boldsymbol{X}_{b3}$ are three agents for deriving $\boldsymbol{Y}_a$, and $\boldsymbol{Z}_a$ is the trial vector for $\boldsymbol{X}_a$. For each $\boldsymbol{X}_a$ in the current population, $\boldsymbol{X}_{b1}$, $\boldsymbol{X}_{b2}$ and $\boldsymbol{X}_{b3}$ are selected from the population in the same manner as the ordinary DE/rand/1/bin.

In the case of the (center, width) model, $Y_{a,i} = (y_{a,i}^c, y_{a,i}^w)$ where,

$$y_{a,i}^c = x_{b1,i}^c + F \cdot (x_{b2,i}^c - x_{b3,i}^c) \quad (9)$$

$$y_{a,i}^w = x_{b1,i}^w + F \cdot (x_{b2,i}^w - x_{b3,i}^w) \quad (10)$$

under the constraint that $y_{a,i}^w \geq 0$. If $y_{a,i}^w$ by (10) becomes smaller than 0, then $y_{a,i}^w$ is repaired as 0.

In the case of the [lower,upper] model, $Y_{a,i} = [y_{a,i}^L, y_{a,i}^U]$ where,

$$y_{a,i}^L = x_{b1,i}^L + F \cdot (x_{b2,i}^L - x_{b3,i}^L) \quad (11)$$

$$y_{a,i}^U = x_{b1,i}^U + F \cdot (x_{b2,i}^U - x_{b3,i}^U) \quad (12)$$

under the constraint that $y_{a,i}^L \leq y_{a,i}^U$. If $y_{a,i}^U$ by (12) becomes smaller than $y_{a,i}^L$ by (11), then the mean value of $y_{a,i}^L$ and $y_{a,i}^U$ is calculated and $y_{a,i}^L$ and $y_{a,i}^U$ are repaired as the mean value.

In (9)-(12), F is the scaling factor which is the same as that in the ordinary DE. With the parent $\boldsymbol{X}_a$ and the donor $\boldsymbol{Y}_a$, the trial vector $\boldsymbol{Z}_a$ is also determined by the same manner as in the ordinary DE: e.g., in the case of employing the binominal crossover, $Z_{a,i}$ is either of $X_{a,i}$ or $Y_{a,i}$ under the crossover rate of CR. With the parent $\boldsymbol{X}_a$ and the trial $\boldsymbol{Z}_a$, the offspring $\boldsymbol{X}_a'$ is also determined by the same manner as in the ordinary DE: the better of $\boldsymbol{X}_a$ or $\boldsymbol{Z}_a$ is employed as $\boldsymbol{X}_a'$.

DE variants other than DE/rand/1/bin can also be extended just in the same manner as described here for DE/rand/1/bin.

IV. Comparison of LU/CW Models for Interval Genotype Values in IDE

As described above, the constraints for the two interval parameters (i.e., the lower and upper values or the center and width values) are different, and thus the methods for modifying constraint-violating values are also different between the LU and CW models. This difference may affect the performance of IDE in searching for solutions. To compare the performances between the two models, IDE with each of the two models is applied to the same task. As the application task, evolution of INNs is employed. The IDE is challenged to evolve INNs which better model a hidden target interval function. The target function is the following interval function in this study, $F(x) = [F(x)^L, F(x)^U]$, where $F(x)^L$ and $F(x)^U$ denote the lower and upper limits of the interval function $F(x)$.

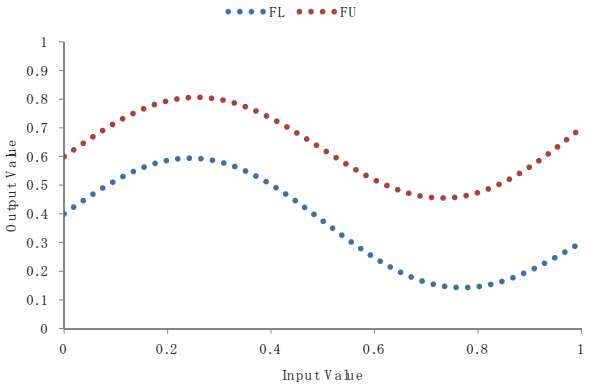

Fig. 3 Target interval function $F(x) = [F(x)^L, F(x)^U]$

$$F(x)^L = 0.2\sin(2\pi x) - 0.1x^2 + 0.4 \quad (13)$$

$$F(x)^U = 0.2\sin(2\pi x) + 0.1x^2 + 0.6 \quad (14)$$

Fig. 3 shows the shape of the target interval function $F(x)$where FL and FU corresponds to $F(x)^L$ and $F(x)^U$ respectively.

The INN is designed as follows.

- #Units: 1 input, 10 hidden, 1 output.
- Unit activation function: the sigmoidal one.

The IDE is designed as follows.

- Initial values for $x_{a,i}^L$, $x_{a,i}^U$, $x_{a,i}^c$: uniformly random within[-1.0, 1.0].
- Initial values for $x_{a,i}^w$: uniformly random within[0.0, 1.0].
- $-10.0 \leq x_{a,i}^L, x_{a,i}^U, x_{a,i}^c \leq 10.0$.
- $0.0 \leq x_{a,i}^w \leq 10.0$.
- #Total INNsevolved in a single run: 1,000,000.
- Population size and #generation: (100,10,000) or (500, 2,000).
- Scaling factor F: 0.5.
- Crossover rate for the binominal crossover CR: 0.8.

The total number of INNs evolved in a single run is set to the same value among the two different population sizes. The number of generations is 10,000 (2,000) for the population size of 100 (500) so that the total number of INNs evolved is constantly 1,000,000 in each run.

Genotype instances $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_P$ are ranked as follows. An INN which corresponds to a genotype instance $\boldsymbol{X}_i$ is supplied with a real input value x_r and calculates its output interval $Y_r = [y_r^L, y_r^U]$. x_r is sampled within the input domain [0, 1] as $x_r = 0.0$, 0.01, 0.02, …, 0.99 and 1.0. Each value of x_r is supplied to the target function $F(x)$ and the interval output value of $F(x_r) = [f_r^L, f_r^U]$ is obtained. Then, the error e_r for x_r is calculated as $e_r = (y_r^L - f_r^L)^2 + (y_r^U - f_r^U)^2$. Fig. 4 illustrates the error between Y_r and $F(x_r)$. e_r is calculated 101 times ($e_0, e_1, \ldots, e_{100}$) for the 101 different values of x_r, and the sum of e_r is used for ranking a genotype instance. A genotype instance with a smaller sum of e_r is ranked better. Note that, in contrast to the training of neural networks by the back propagation algorithm, the error scores are *not* utilized for calculating the modification amount of weight/bias values. The error scores are utilized for only ranking the genotype instances. Thus, the target function $F(x)$ is completely hidden from the reproduction process of the IDE.

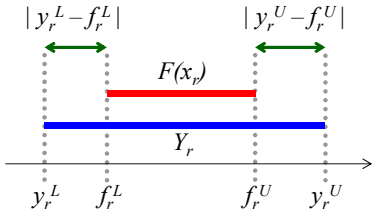

Fig. 4 Error between the target interval $Y_r = [y_r^L, y_r^U]$ and the INN output $F(x_r) = [f_r^L, f_r^U]$ for the input value x_r.

Fig. 5 shows the output interval function of the best INN among the total 20,000,000 INNs (= [1,000,000 INN in each run] ∗ [5 runs] ∗ [2 variations of population sizes] ∗ [2 variations of interval models]) evolved by the IDE. The best INN could model the target function $F(x)$ with the error score of $1.5 * 10^{-3}$. The best INN could model the target function $F(x)$ very well.

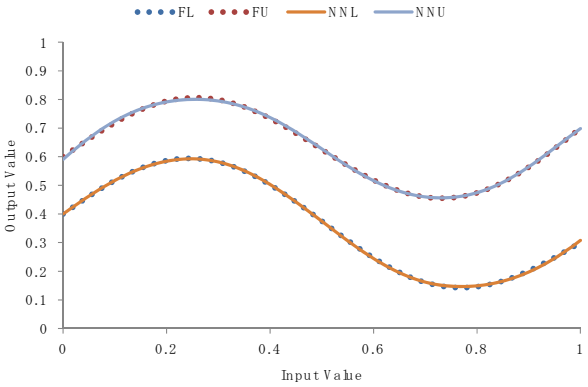

Fig. 5 Output interval function of the best INN evolved by IDE

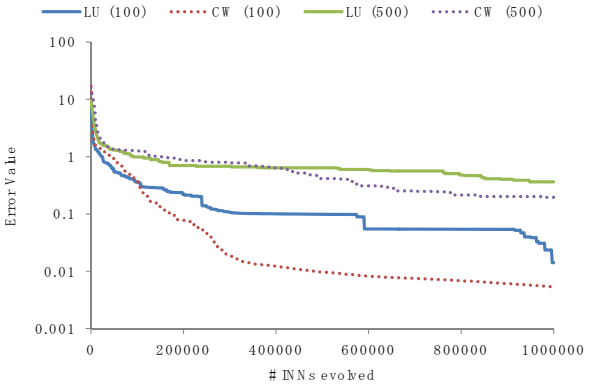

Fig. 6 Error values of the best INN at each number of INNs evolved (the error values are averaged over 5 runs)

Fig. 6 shows the error values of the best INN among each number of INNs evolved (e.g., 500,000 INNs are evolved in total at the 5,000th generation with the population size of 100). In Fig. 6, “LU (100)” denotes the result with the LU 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 5 runs. Fig. 6 revealed that the CW model contributed better to the IDE than the LU model did with both of the population sizes of 100 and 500. After the 1,000,000 INNs evolved, the dotted curves for the CW model went below the solid curves for the LU model.

V. Conclusion

The two models for specifying intervals, i.e., the LU model and the CW model, were compared so that which model was better for the genotype values in the interval-valued differential evolution. The experimental result indicates that the CW model is better than the LU model. In future work, the author will further compare the two models for other application tasks and confirm that the CW model contributes better than the LU model. In addition, the author will investigate the reason why the model contributes better.

Acknowledgment

This work is supported by Kyoto Sangyo University Research Grant.

References

[1] R. Storn and K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, Vol.11, No.4, pp.341-359, 1997.

[2] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford Univ. Press, 1996.

[3] H. Okada, Proposal of fuzzy evolutionary algorithms for fuzzy-valued genotypes, *Proc. of Int. Conf. on Instrumentation, Control, Information Technology and System Integration (SICE Annual Conference) 2012*, pp.1538-1541, 2012.

[4] H. Ishibuchi, H. Tanaka and H. Okada, An architecture of neural networks with interval weights and its application to fuzzy regression analysis, *Fuzzy Sets and Systems*, Vol.57, No.1, pp.27-39, 1993.

[5] H. Okada: Interval-valued differential evolution for evolving neural networks with interval weights and biases, *Proc. of the 6th International Workshop on Computational Intelligence & Applications (IWCIA2013)*, pp.81-84, 2013.

[6] G. Alefeld and J. Herzberger, *Introduction to Interval Computation*, Academic Press, 1983.