Journal of Korean Institute of Intelligent Systems, Vol. 22, No. 4, August 2012, pp. 399-404
http://dx.doi.org/10.5391/JKIIS.2012.22.4.399

개선된 미분 진화 알고리즘에 의한 퍼지 모델의 설계

Design of Fuzzy Models with the Aid of an Improved Differential Evolution

김현기 · 오성권[†]
Hyun-Ki Kim, and Sung-Kwun Oh[†]

수원대학교 전기공학과

Abstract

Evolutionary algorithms such as genetic algorithm (GA) have been proven their effectiveness when applying to the design of fuzzy models. However, it tends to suffer from computationally expensWive due to the slow convergence speed. In this study, we propose an approach to develop fuzzy models by means of an improved differential evolution (IDE) to overcome this limitation. The improved differential evolution (IDE) is realized by means of an orthogonal approach and differential evolution. With the invoking orthogonal method, the IDE can search the solution space more efficiently. In the design of fuzzy models, we concern two mechanisms, namely structure identification and parameter estimation. The structure identification is supported by the IDE and C-Means while the parameter estimation is realized via IDE and a standard least square error method. Experimental studies demonstrate that the proposed model leads to improved performance. The proposed model is also contrasted with the quality of some fuzzy models already reported in the literature.

Key Words : Improved Differential Evolution (IDE), Fuzzy Inference System (FIS), Information Granulation (IG), C-Means clustering, Least Square Method (LSM)

1. Introduction

A great deal of development of fuzzy modeling has been reported over the past decades. In these publications, pioneering works such as Tong et al. [1] are well-known due to different approaches for fuzzy modeling. In these studies, an important problem of designing a fuzzy model is how to identify “good” initial parameters of the fuzzy rules. To address this problem, Oh et.al [2] have presented that using genetic algorithm and a concept of Information granulation (IG) to develop fuzzy inference systems. Liu et.al [3], Chung and Kim [4] and others have discussed the design of fuzzy models by means of optimization algorithms, respectively. Some enhancements have been obtained, yet there is a lack of investigations on the solution space being explored and studies on ways in which the computational effectiveness could be enhanced.

As one of successful optimization vehicle, genetic algorithm (GA) is widely applied to optimize fuzzy inference system. GA is a global search technique with the ability to explore a large space for suitable solutions only requiring a performance measure. In addition to their ability to find near optimal solutions in complex search spaces, the genetic code structure and independent performance features of GA make it suitable candidates to incorporate a priori knowledge [5]. In spite of these advantages, GA is also has its limitations. For example, the convergence speed is sometimes computationally expensive due to its evolutionary process. Differential Evolution (DE) initialized by Price and Storn in 1995 [6] is an effective global optimization algorithm which has fast convergence speed. Nevertheless, an apparent limitation is that DE tends to suffer from premature convergence when solving complex problems.

Recently, an improved differential evolution (IDE) has been proposed [7]. The IDE has been proven that it can search the solution space more efficiently and it is well suited for parallel implementation. Here we employ the IDE as an optimization vehicle to optimize the fuzzy models. A design of fuzzy inference systems based on the IDE is developed. The evaluation of the performance of the proposed model is carried out by using two

접수일자: 2012년 4월 20일
심사(수정)일자: 2012년 4월 20일
게재확정일자 : 2012년 6월 27일
[†] 교신 저자
"본 논문은 본 학회 2012년도 춘계학술대회에서 선정된 우수논문입니다."
본 논문은 경기도의 경기도지역협력연구센터사업의 일환으로 수행하였음[GRRC 수원2012-B2, U-city 보안감시 기술협력센터], 그리고 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2012-003568)

well-known data sets. To demonstrate the performance of IDE, the proposed model is also compared with some existing fuzzy models that have already reported in some publications.

The paper is organized as follows. In Section 2, we first review the main different DE and then introduce the framework of IDE in detail. Section 3 gives the design of fuzzy models. Section 4 reports a series of comparative studies by using different well-known datasets. Finally, the conclusion is covered in Section 5.

2. Improved Differential Evolution (IDE)

Difference evolution (DE) has been successfully applied to solve lots of different real-world optimization problems. As one of evolutionary algorithms, there also three basic operators in DE, they are Mutation, Crossover, and Selection. However, DE has been shown to have certain weaknesses, especially if the global optimum should be located using a limited number of function evaluations [7]. To overcome this drawback, orthogonal approach is included to improve the initialization as well as crossover. The IDE is designed with the aid of an orthogonal design method, which can accelerate its convergence rate. In the IDE, the orthogonal design method is not only to be used to generate the initial population, but also to be applied to design the crossover operator. Table 1 summarizes the pseudo code of IDE.

Table 1. Pseudo of IDE

```
/* Pseudo of IDE*/
BEGIN
   Initialization   with the aid of orthogonal;
   While the   terminal condition is satisfied
       Selection;
       Mutation;
       Orthogonal   crossover;
       Evaluation;
   End While
   Report the   result;
End
```

1) Initialization. The initialization population should be better cover the entire solution space as much as possible by uniformly randomizing individuals. Generally, if the solution space is large, a big population size is needed for DE. It is apparent that it is desirable to sample a small, but representative population of individuals for experimentation, and based on the sample, the optimal may be estimated. The orthogonal design was developed for the purpose [8]. The main steps of orthogonal population are as follows:

Step 1. Design an orthogonal array.
Step 2. Quantize the solution space.
Step 3. Generate the population.

In a discrete single-objective optimization problem, when there are N factors (variables) and each factor has Q levels, the search space consists of QN combinations of levels. When N and Q are large, it may not be possible to do all QN experiments to obtain optimal solutions. For convenience, we use L(Qc)todenotetheorthogonal array with different level Q, where Q is odd. R that is equal to Qj is used to indicate the number of orthogonal array, where j is a positive integer fulfilling

$$C=\frac{Q^{J}-1}{Q-1} \qquad (1)$$

Here C denotes the number of the columns of orthogonal array in the above equation. More specifically, the algorithm of orthogonal initialization is listed as shown in Table 2.

Table 2. Pseudo of orthogonal initialization

```
/* Algorithm 1. orthogonal initialization*/
BEGIN
   Eval=0;
   For i=1 to R do
     For j=1 to N do
        Generate orthogonal array.
        Evaluation;
   End For /*i*/
   End For /*j*/
End
```

2) Mutation. After initialization, DE employs the mutation operation to produce a mutant vector with respect to each individual, so-called target vector, in the current population. There are several mutation strategies implemented in the DE codes [9]. The five most frequently used mutations operations are listed as follows:

1)"DE/rand/1"

$$X_{new}=X_1+F\cdot(X_2-X_3) \qquad (2)$$

2)"DE/best/1"

$$X_{new}=X_{best}+F\cdot(X_1-X_2) \qquad (3)$$

3)"DE/best/2"

$$X_{new}=X_{best}+F\cdot(X_1-X_2)+F\cdot(X_3-X_4) \qquad (4)$$

4)"DE/best/1"

$$X_{new}=X_1+F\cdot(X_2-X_3)+F\cdot(X_4-X_5) \qquad (5)$$

Where, X_1, X_2, X_3, X_4, X_5 are individuals randomly selected from the population, X_{best} is the individual with the best fitness in the current population, F is the scaling factor that is a positive control parameter for scaling the difference vector.

3) Crossover. Generally, the binomial (uniform) crossover of DE can be briefly stated as follows:

$$V_i = \begin{cases} V_i, & if\,(rand[0,1] \le CR); \\ X_i, & otherwise \end{cases} \tag{6}$$

Where, CR denotes the crossover rate that is a user-specified constant within the range [0,1), which controls the fraction of parameter values copied from the mutant vector [9]. Based on this crossover operation, the orthogonal crossover operation can be developed as shown in Table 3 [7].

Table 3. Pseudo of orthogonal crossover

```
/* Algorithm 2. orthogonal initialization*/
BEGIN
    Set two parameters $Q_1$and$C_1$;
    Select the smallest J1 fulfilling   $(Q_{J1}-1)/(Q_1-1)=C_1$
    Randomly select two solutions from   the population
    Quantize the domain formed by the   two solutions
    Generate $R_1$potentialoffspringbasedonQuantization
    Select the best solution B from $R_1$offspring
    Output the best solution B
End
```

4) Selection. The selection operation of IDE is the same as the conventional DE. If the values of some parameters of a newly generated trial vector exceed the corresponding upper and lower bounds, we randomly and uniformly reinitialize them within the prespecified range. Then the objective function operation is performed and a selection operation is performed. The selection of DE can be briefly stated as follows [9] :

$$V_i = \begin{cases} X_i, & if\;\; f(X_i) \le f(V_i); \\ V_i, & otherwise \end{cases} \tag{7}$$

3. Design of the IG-based fuzzy models

With respect to the design of fuzzy models, we realized the structure identification as well as parameter identification. The structure identification is supported by the IDE and C-means while the parameter estimation is realized via the IDE and weighted least square error method. The identification of the conclusion parts of the rules deals with a selection of their structure (type 1, type 2, type 3 and type 4) that is followed by the determination of the respective parameters of the local functions occurring there. The conclusion part of the rule that is extended form of a typical fuzzy rule in the TSK (Takagi-Sugeno-Kang) fuzzy model has the form.

$$If\, x_i\, is\, A_{1c}\, \text{and} \cdots \text{and} x_k\, is\, A_{kc}\, then\, y_j - M_j = f_j(x_1, \cdots x_k) \tag{8}$$

Type 1 (Simplified Inference):

$$f_j = a_{j0} \tag{9}$$

Type 2 (Linear Inference):

$$f_j = a_{j0} + a_{j1}(x_1 - V_{j1}) + \cdots + a_{jk}(x_k - V_{jk}) \tag{10}$$

Type 3 (Quadratic Inference):

$$\begin{aligned} f_j &= a_{j0} + a_{j1}(x_1 - V_{j1}) + \cdots + a_{jk}(x_k - V_{jk}) \\ &+ a_{j(k+1)}(x_1 - V_{1j})^2 + \cdots + a_{j(2k)}(x_k - V_{jk})^2 \\ &+ a_{j(2k+1)}(x_1 - V_{1j})(x_2 - V_{2j}) + \cdots \\ &+ a_{j((k+2)(k+1)/2)}(x_{k-1} - V_{(k-1)j})(x_k - V_{kj}) \end{aligned} \tag{11}$$

Type 4 (Modified Quadratic Inference):

$$\begin{aligned} f_j &= a_{j0} + a_{j1}(x_1 - V_{j1}) + \cdots + a_{jk}(x_k - V_{jk}) \\ &+ a_{j(k+1)}(x_1 - V_{1j})(x_2 - V_{2j}) + \cdots \\ &+ a_{j(k(k+1)/2)}(x_{k-1} - V_{(k-1)j})(x_k - V_{kj}) \end{aligned} \tag{12}$$

The optimal coefficients of the model is estimated through the minimization of the objective function J_L

$$J_L = \sum_{i=1}^{n} \sum_{k=1}^{m} w_{ik}(y_k - f_i(x_k - v_i))^2 \tag{13}$$

Where w_{ik} is the normalized firing strength (activation level) of the ith rule.

The performance index J_L can be rearranged as

$$\begin{aligned} J_L &= \sum_{i=1}^{n} (Y - X_i a_i)^T W_i (Y - X_i a_i) \\ &= \sum_{i=1}^{n} (W_i^{1/2} Y - W_i^{1/2} X_i a_i)^T (W_i^{1/2} Y - W_i^{1/2} X_i a_i) \end{aligned} \tag{14}$$

Where a_i is the vector of coefficients of ith consequent polynomial (local model), Y is the vector of output data, W_i is the diagonal matrix (weighting factor matrix) which represents degree of activation of the individual information granules by the input data. X_i is a matrix which is formed with input data and information granules (centers of cluster). In case the consequent polynomial is Type 2 (linear or a first-order polynomial), X_i and a_i read as follows

$$W_i = \begin{bmatrix} W_{i1} & 0 & \cdots & 0 \\ 0 & W_{i2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W_{im} \end{bmatrix} \tag{15}$$

$$X_i=\begin{bmatrix}1\ (x_{11}-v_{i1})\cdots(x_{l1}-v_{il})\\ 1\ (x_{12}-v_{i1})\cdots(x_{l2}-v_{il})\\ 1\quad \vdots \quad \ddots \quad \vdots\\ 1\ (x_{1m}-v_{i1})\cdots(x_{lm}-v_{il})\end{bmatrix}\tag{16}$$

$$a_i=[a_{i0}\,a_{i1}\cdots a_{il}]\tag{17}$$

For the local learning algorithm, the objective function is defined as a linear combination of the squared error, which is a difference between output data and the result produced each fuzzy rule when considering the weighting factor matrix W_i. This matrix captures the activation levels of input data with respect to ith sub-space. In this sense we can consider the weighting factor matrix as a discrete version of the fuzzy (linguistic) representation for the corresponding sub-space. The optimal coefficients of the consequent polynomial of the ith fuzzy rule can be determined in a usual manner that is

$$a_i=(X_i^T W_i X_i)^{-1} X_i W_i Y\tag{18}$$

Notice that the coefficients of the consequent polynomial of each fuzzy rule have been computed independently using a subset of training data. These computations can be implemented in parallel and in this case the overall computing load becomes unaffected by the total number of the rules.

The proposed IDE is exploited here to optimize the fuzzy models. Here we simultaneously realize the structure identification as well as parameter estimation of the model [10-11]. A stochastic variable (a variant identification ratio) used within modified simple differential evolution operators (crossover and mutation) in the IDE is used support an efficient successive tuning method. During the initial generations of the IDE, the differential evolution operators are assigned with higher probability to the solution region for structural optimization. This probability becomes lower when dealing with a region of the solution involving the parameter estimation. In this manner, the optimization becomes mostly focused on the structure identification. Over the course of the optimization (for higher generations), the optimization of the fuzzy model becomes predominantly focused on the parameter estimation.

The differential evolution operators in the IDE for the successive tuning method being realized with the aid of a variant identification ratio are implemented. Their essential parameters such as gen, maxgen, and l are given. Here, gen is an index of the current generation, maxgen stands for the maximal number of generations being used in the algorithm, and l serves as some adjustment coefficient whose values can determine a variant identification ratio (p) for both structure identification and parameter estimation. The detailed space search operator in the IDE algorithm is presented as follows:

```
While { the termination conditions are not met }
Generate random variable (r1).
Calculate a variant identification ratio (p) which is a generation-based stochastic variable of the form
```

$$p=\frac{r_1+(1-\geq n/\max\geq n)}{\lambda}\tag{19}$$

```
IF {p > 0.5}
Differential evolution within the first part of solutions for structural optimization.
Else
Differential evolution within the second part of solutions for parametric optimization.
End IF
End while
```

case the overall computing load becomes unaffected by the total number of the rules.

The objective function (performance index) is regarded as a basic mechanism guiding the evolutionary search carried out in the solution space of potential solutions. The objective function involves both the training and testing data and comes as a convex combination of these two components

$$MPI=\theta\times PI+(1-\theta)\times V\_PI\tag{20}$$

Here, PI and V_PI denote the performance index for the training data and validation data, respectively. θ is a weighting factor that allows us to form a sound balance between the performance of the model for the training and testing data. Depending upon the values of the weighting factor, several specific cases of the objective function are worth distinguishing.

(i) If $\theta=1$ then the model is optimized based on the training data. No testing data is taken into consideration.

(ii) If $\theta=0.5$ then both the training and testing data are taken into account. Moreover it is assumed that they exhibit the same impact on the performance of the model.

(iii) The case $\theta=\alpha$ where $\alpha\in[0,1]$ embraces both the cases stated above. The choice of α establishes a certain tradeoff between the approximation and generalization aspects of the fuzzy model.

We use performance index of the standard root mean squared error (RMSE) and mean squared error (MSE)

$$MPI(\text{or } E\_PI)=\begin{cases}\sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i-y_i^*)^2}, & (\text{RMSE})\\ \frac{1}{m}\sum_{i=1}^{m}(y_i-y_i^*)^2 & (\text{MSE})\end{cases}\tag{21}$$

Where, E_PI represents as the performance index of testing data.

4. Experimental studies

This section reports on comprehensive numeric studies illustrating the design of the fuzzy model. We use two well-known data sets. Each data set is divided into two parts of the same size. PI, V_PI, and E_PI denote the performance index, respectively. Where, PI is for the training data, V_PI represents the validation data, and E_PI stands for the testing data. In all considerations, the weighting factor θ was set to 0.5.

4.1 Automobile MPG Data

The first dataset is an automobile MPG data (ftp://ics.uci.edu/pub/machine-learning-data-based/auto-mpg) with the output being the automobile's fuel consumption expressed in miles per gallon. This data set includes 392 input-output pairs (after removing incomplete instances) where the input space involves 8 input variables. To come up with a quantitative evaluation of the fuzzy model, we use the standard RMSE performance index. The automobile MPG data is partitioned into two separate parts. The first 196 data pairs are used as the training data set for FIS, the second 118 pairs are utilized for validation data set, while the remaining 78 pairs are the testing data set for assessing the predictive performance. The identification error of the proposed model is compared with the performance of some other model; refer to Table 4. It is clear that IDE-based fuzzy model leads to better performance in comparison with some other models.

Table 4. Comparative analysis of selected models(MPG)

Model	PI	V_PI	E_PI	No.of rules
RBFNN [12]	3.24		3.62	36
Linguistic model [13]	2.86		3.24	36
Functional RBFNN[12]	2.41		2.82	33
Our model	2.05	2.69	2.54	32

4.2 Medical Image System Data (MIS)

The second data we consider a medical imaging system data set involves 390 software modules written in Pascal and FORTRAN. Each module is described by 11 input variables, that is, total lines of code including comments (LOC), total code lines (CL), total character count (TChar), total comments (TComm), number of comment characters (MChar), number of code characters (DChar), Halstead's program length (N), Halstead's estimated program length (N), Jensen's estimator of program length (NF), McCabe's cyclomatic complexity (V(G)), and Belady's bandwidth metric (BW). The output variable of the model is the number of reported changes－change reports (CRs). The given dataset is partitioned to producet hree data sets. The first 50% of dataset is used for training the models, the second 30% of dataset is utilized fo rvalidation, while the remaining 20% of dataset provides for the testing dataset. RMSE is considered as the performance index.

For the MIS dataset, the optimal performance of the proposed fuzzy model is summarized in Table 5. As shown in the results, the performances of the proposed fuzzy model are better in the sense of its approximation and prediction abilities. That is to say, the proposed fuzzy model leads to stable good performance when considering a sound balance between the approximation and generalization.

Table 5. Comparative analysis of selected models (MIS)

Model	PI	VPI	EPI	Index
SONFN[14]	35.745		17.807	MSE
FPNN[15]	32.195		18.462	MSE
FSONN[16]	23.739		9.090	MSE
Incremental model[17]	5.877		6.570	RMSE
Our Model	0.671	1.186	1.912	RMSE

5. Conclusion

This paper introduced an approach to develop the fuzzy models by means of an IDE. Experimental results show that IDE-based model obtains better performance in comparison with some other fuzzy models reported in the literature.

Reference

[1] R.M. Tong, "The evaluation of fuzzy models derived from experimental data," *J. Fuzzy Sets and Syst.,* vol. 13, pp. 1-12, 1980.

[2] C.W. Xu, Y. Zailu, "Fuzzy model identification self learning for dynamic system," *J. IEEE Trans. on System, Man, and Cybernetics*, vol. 17, no. 4, pp. 683-689, 1987.

[3] S.K. Oh., W. Pedrycz, "Identification of Fuzzy Systems by means of an Auto-Tuning Algorithm and Its Application to Nonlinear Systems," *J. Fuzzy Sets and Syst.*, vol. 115, no. 2, pp. 205-230, 2000.

[4] S.K. Oh, W. Pedrycz, K.J Park, “Identification of fuzzy systems by means of genetic optimization and data granulation,” *Journal of Intelligent & Fuzzy Systems*, vol 18, pp. 31-41, 2007.

[5] F. Herrera, “Genetic fuzzy stystems: taxonomy, current research trends, and prospects,” *Evol. Intel.*, vol 1, pp. 27-46, 2008.

[6] R. Storn, K. V. Price, “Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces,” *Global Optim.*, vol. 11, pp. 341-359, 1997.

[7] G. Wenyin, C. Zhihua, J. Liangxiao, “Enhancing the performance of differential evolution using orthogonal design method,” *Applied Mathematics and Computation*, vol. 206, pp. 56-69, 2008.

[8] Q.Zhang, Y.W.Leung, “Orthogonal genetic algorithm for multimedia multicast routing,” *IEEE Transactions on Evolutionary Computation,* vol. 3, pp. 53-62, 1999.

[9] A.K. Qin, V.L.Huang, P.N. Suganthan. “Differential evolution algorithm with strategy adaptation for global numerical optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 13, pp. 398-417, 2009.

[10] W. Huang, L. Ding, S.K. Oh, C.W. Jeong, and S.C. Joo. “Identification of fuzzy inference system based on information granulation,” *KSII Transactions on Internet and Information systems*, vol. 4, pp. 575-594, 2010.

[11] W. Huang, S.K. Oh, L. Ding, H.K. Kim, S.C. Joo, “Identification of fuzzy inference systems using a multi-objective space search algorithm and in formation granulation,” *Journal of Electrical Engineering & Technology*, vol. 6, pp. 853-866, 2011.

[12] Pedrycz, W., and Kwak, K.C. “Linguistic models as a framework of user-centric system modeling,” *IEEE Trans. Syst., man cybern. - PART A: Systems and humans*, vol. 36, no. 4, pp. 727 - 745, 2006.

[13] Pedrycz, W., Park, H.S., and S.K. Oh, “A granular-oriented development of functional radial basis function neural networks,” *Neurocomputing*, vol. 72, pp. 420 - 435, 2008.

[14] S. K. Oh, W. Pedrycz, and B. J. Park, “Relation-based Neurofuzzy Networks with Evolutionary Data Granulation,” *Mathematical and Computer Modeling*, vol. 40, no. 7-8, pp. 891-921, 2004.

[15] S. K. Oh and W. Pedrycz, “Fuzzy Polynomial Neuron-Based Self-Organizing Neural Networks,” *Int. J. of General Systems*, vol. 32, no. 3, pp. 237-250, 2003.

[16] S. K. Oh, H. S. Park, C. W. Jeong, and S. C. Joo, “GA-based Feed-forward Self-organizing Neural Network Architecture and Its Applications for Multi-variable Nonlinear Process Systems,” *KSII Transactions on Internet and Information Systems,* vol. 3, no. 3, pp. 309-330, Jun, 2009.

[17] W. Pedrycz and K. C. Kwak, “The Development of Incremental Models,” *IEEE Trans. Fuzzy Systems*, vol. 15, no. 3, pp. 507-518, 2007.

저자소개

김현기(Hyun-Ki Kim)

1977년 : 연세대학교 전기공학과 공학사
1985년 :연세대학교 전기공학과 공학석사
1991년 : 연세대학교 전기공학과 공학박사

1997년~2010년 : 경기지역 산학연 기술지원단 단장
1998년~2000년 :사업기술정책연구소 심의위원
2000년~2004년 :사단법인 산학연 컨소시엄 적국협의회회장
1989년~현재 : 수원대학교 전기공학과 교수
관심분야 : Automatic System and Control, Intelligent Modeling and Control.
Phone : 031-220-2666
E-mail : hkkim@suwon.ac.kr

오성권(Sung-Kwun Oh)

1981년 : 연세대학교 전기공학과 졸업
1983년 : 동 대학원 석사
1993년 : 동 대학원 박사
1983-1989 : 금성산전연구소(선임연구원).
1996-1997 : 캐나다 Manitoba 대학 전기 및 컴퓨터공학과 Post-Doc.
1993-2004 : 원광대학교 전기전자 및 정보공학부 교수.
2005~현재 : 수원대학교 전기공학과 교수, 2002~현재 대한전기학회, 제어로봇시스템학회, 퍼지 및 지능시스템학회 편집위원.
관심분야 : 퍼지시스템, 퍼지-뉴럴네트워크, 자동화 시스템, 고급 computational intelligence, 지능제어 등.
Phone : 031-229-8162
E-mail : ohsk@suwon.ac.kr